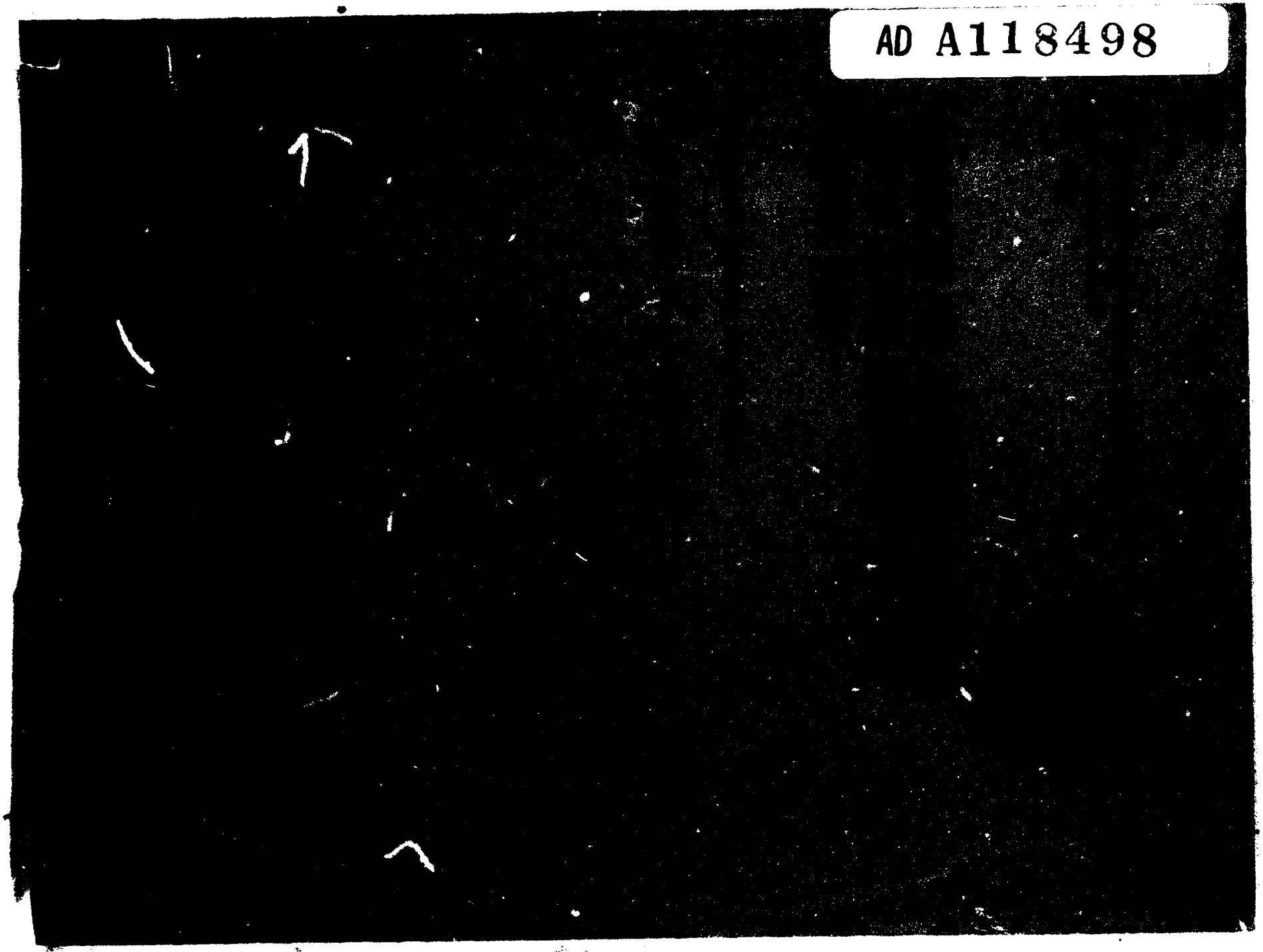
AD-A118 498 MCDONNELL DOUGLAS RESEARCH LABS ST LOUIS MO F/G 9/2
ALGORITHM FOR SURFACE OF TRANSLATION ATTACHED RADIATORS (A-STAR--ETC(U)
MAY 82 L N MEDGYESI-MITSCHANG, J M PUTNAM F30602-80-C-0106
UNCLASSIFIED RADC-TR-82-113-VOL-3 NL

1 of 3
AD
A118498



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: RADC-TR-82-113, Vol III (of three)	2. GOVT ACCESSION NO.: AD-A118 498
	3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): ALGORITHM FOR SURFACE OF TRANSLATION ATTACHED RADIATORS (A-STAR): Computer Codes	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report 17 Mar 80 - 16 Oct 81
	6. PERFORMING ORG. REPORT NUMBER: N/A
7. AUTHOR(s): L. N. Medgyesi-Mitschang, J. M. Putnam	8. CONTRACT OR GRANT NUMBER(s): F30602-80-C-0106
9. PERFORMING ORGANIZATION NAME AND ADDRESS: McDonnell Douglas Research Laboratories, McDonnell Douglas Corporation, St Louis MO 63166	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 64747F, 20640309
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (BRCC), Griffiss AFB NY 13441	12. REPORT DATE: May 1982
	13. NUMBER OF PAGES: 245
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Same	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Daniel E. Warren (RBCC)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Radiation
Scattering
Conducting asymmetric surfaces
Electromagnetic Compatibility
Method of moments
Near field analysis
Antenna coupling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Listing of the computer codes implementing the algorithm for radiation and scattering from bodies of translation, described in Volumes I and II, is given.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

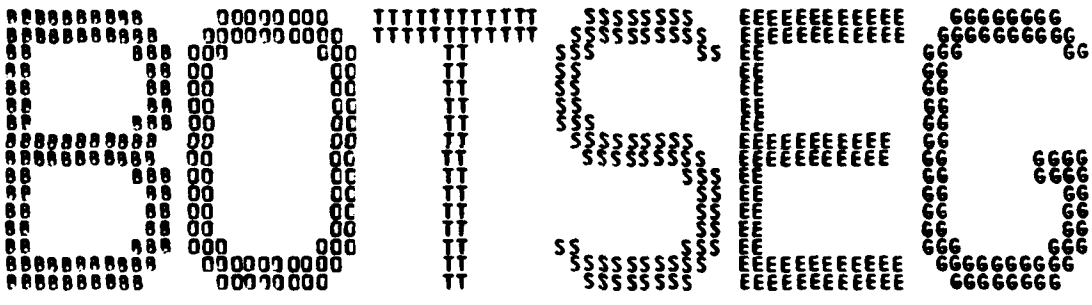
UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

	Page No.
LISTING OF CODES	1
BOTSEG	1
BOTZSS	7
BOTZSC	21
BOTZCC	35
BOTZSW	53
BOTZWW	69
BOTZCW	85
BOTINV	101
BOTINVA	113
BOTAC	129
BOTRA	147
BOTSCM	187
BOTSCB	211

LISTING OF CODES

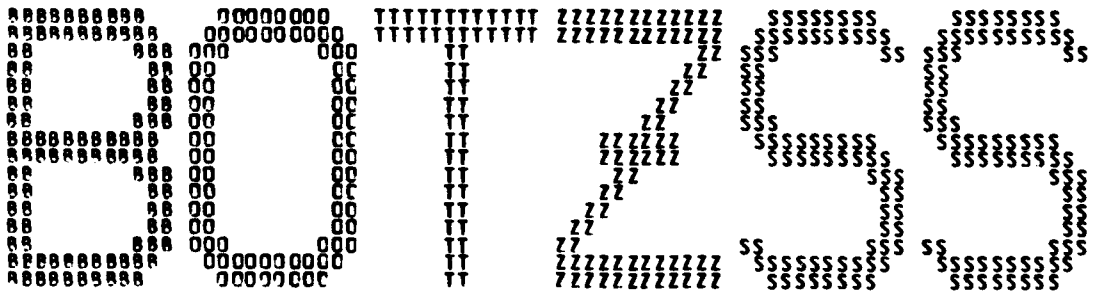


```
      REAL XB(83),YB(83)
      REAL XTAB(100),YTAB(100),XC(100),YC(100)
      READ(5,2) NPTS,NP
    2 FORMAT(2I3)
C
C READ BODY COORDINATES.
C
      DO 100 I=1,NPTS
      XC(I)=0.0
      YC(I)=0.0
  100 READ(5,3) XTAB(I),YTAB(I),XC(I),YC(I)
    3 FORMAT(4E10.4)
C
C SEGMENT THE BODY INTO NP-1 SEGMENTS OF EQUAL LENGTH.
C
      CALL INTRPL(NP-1,NPTS,P,XTAB,YTAB,XC,YC,XB,YB)
      WRITE(6,1) P,NP
    1 FORMAT(//,12H PERIMETER =,F10.4,///,
     1 5H NP =,I4)
      WRITE(6,53) (YB(I),I=1,NP)
      WRITE(6,54) (XB(I),I=1,NP)
   53 FORMAT(//,3H YB,/,(1X,10F10.4))
   54 FORMAT(//,3H XB,/,(1X,10F10.4))
C
C PLOT THE CALCULATED BODY COORDINATES.
C
      CALL PLOTB(XB,YB,NP,31)
      STOP
      END
C*********************************************************************
      SUBROUTINE INTRPL(NI,NP,P,XTAB,YTAB,XC,YC,XR,YR)
C
C     THIS SUBROUTINE WILL TAKE BODY COORDINATES (XTAB(I),YTAB(I),I=1,NP)
C    AND CALCULATE EQUALLY SPACED BODY COORDINATES (XR(I),YR(I),I=1,NI+1)
C     THE POINTS XTAB AND YTAB ARE ASSUMED TO BE EITHER
C    1) ENDPOINTS OF STRAIGHT LINES OR 2) ENDPOINTS OF CIRCULAR ARCS.
C     FOR EXAMPLE IF EITHER XC(K) OR YC(K) IS NOT ZERO, THEN
C    (XTAB(K),YTAB(K)) AND (XTAB(K+1),YTAB(K+1)) ARE ENDPOINTS OF THE
C    CIRCULAR ARC WITH CENTER (XC(K),YC(K)).  NOTE, IF THE RADIUS
C    IS DIFFERENT AT THE ENDPOINTS, IT IS ASSUMED TO HAVE A LINEAR
C    DEPENDENCE WITH ANGLE BETWEEN THE ENDPOINTS.  IF XC(K)=YC(K)=0,
C    IT IS ASSUMED TO BE STRAIGHT.
C     P IS RETURNED AS THE PERIMETER OF THE BODY.
C
      REAL XTAB(1),YTAB(1),XR(1),YR(1)
      REAL XC(1),YC(1)
      REAL MOVE
      P=0.0
      K=NP-1
      DO 25 I=1,K
      IF(XC(I).NE.0.0 .OR. YC(I).NE.0.0) GO TO 20
      X=XTAB(I+1)-XTAB(I)
```

```
C2060         Y=YTAB(I+1)-YTAB(I)
C2070         P=P+SQRT(X*X+Y*Y)
C2080         GO TO 25
C2090   20    X=XTAB(I)-XC(I)
C2100         Y=YTAB(I)-YC(I)
C2110         R=SQRT(X*X+Y*Y)
C2120         X1=XTAB(I+1)-XC(I)
C2130         Y1=YTAB(I+1)-YC(I)
C2140         R1=SQRT(X1*X1+Y1*Y1)
C2150         THETA=ACOS((X*X1+Y*Y1)/(R*R1))
C2160         P=P+(R+R1)*THETA/2.0
C2170   25    CONTINUE
C2180         PI=P/NI
C2190         XR(1)=XTAB(1)
C2200         YR(1)=YTAB(1)
C2210         XR(NI+1)=XTAB(NP)
C2220         YR(NI+1)=YTAB(NP)
C2230         NK=1
C2240         MOVE=0.0
C2250         DO 200 I=1,K
C2260         XI=XTAB(I)
C2270         YI=YTAB(I)
C2280         IF(XC(I).NE.0.0 .OR. YC(I).NE.0.0) GO TO 40
C2290         X=XTAB(I+1)-XI
C2300         Y=YTAB(I+1)-YI
C2310         D=SQRT(X*X+Y*Y)
C2320         GO TO 50
C2330   40    X=XI-XC(I)
C2340         Y=YI-YC(I)
C2350         R=SQRT(X*X+Y*Y)
C2360         X1=XTAB(I+1)-XC(I)
C2370         Y1=YTAB(I+1)-YC(I)
C2380         R1=SQRT(X1*X1+Y1*Y1)
C2390         THETA=ACOS((X*X1+Y*Y1)/(R*R1))
C2400         D=(R+R1)*THETA/2.0
C2402         THETA=ATAN2(Y,X)
C2403         THETA1=ATAN2(Y1,X1)
C2404C  DIR= +1 FOR CLOCKWISE DIRECTION
C2405C       -1 FOR COUNTER-CLOCKWISE DIRECTION.
C2406         DIR=+1.
C2407         IF(THETA.GT.THETA1 .AND. THETA-THETA1.GT.3.14159266) DIR=-1.
C2408         IF(THETA.LT.THETA1 .AND. THETA1-THETA.LT.3.14159265) DIR=-1.
C2410   50    IF(MOVE+D.LT.PI) GO TO 200
C2420         NK=NK+1
C2430         IF(XC(I).NE.0.0 .OR. YC(I).NE.0.0) GO TO 60
C2440         XR(NK)=XI+(PI-MOVE)*(XTAB(I+1)-XI)/D
C2450         YR(NK)=YI+(PI-MOVE)*(YTAB(I+1)-YI)/D
C2460         GO TO 70
C2470   60    X=XI-XC(I)
C2480         Y=YI-YC(I)
C2490         R=SQRT(X*X+Y*Y)
C2500         THETA=ATAN2(Y,X)
```

```
C2510       R=R+(PI-MOVE)*(R1-R)/D
C2520       XR(NK)=XC(I)+R*COS(THETA-DIR*(PI-MOVE)/R)
C2530       YR(NK)=YC(I)+R*SIN(THETA-DIR*(PI-MOVE)/R)
C2540  7C   IF(NK.EQ.NI) RETURN
C2550       D=D-PI+MOVE
C2560       MOVE=C.O
C2570       XI=XR(NK)
C2580       YI=YR(NK)
C2590       GO TO 50
C2600  2CC  MOVE=MOVE+D
C2610       RETURN
C2620       END
C2910C ***************************************************
C2920       SUBRCUTINE PLOTB(X,Y,N,NR)
C2930C ***************************************************************
C2940C
C2950C WRITTEN 2/14/74   BY J. M. PUTNAM       DEPT 220     X23877
C2960C
C2970C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C2980C    N IS THE NUMBER OF POINTS TO BE PLOTTED.
C2990C    NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C3000C    NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C3010C       NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C3020C
C3030       REAL X(1),Y(1),HEAD(10)
C3040       INTEGER  LINE(101),BLANK,STAR
C3050       DATA BLANK,STAR /1H ,1H*/
C3060       NC=51
C3070       N10=(NC-1)/10
C3080       WRITE(108,500)
C3090  5CC  FORMAT(//,17H BODY COORDINATES)
C3100       WRITE(108,504)
C3110       XMIN=X(1)
C3120       XMAX=X(1)
C3130       YMIN=Y(1)
C3140       YMAX=Y(1)
C3150       DO 6 I=1,N
C3160       IF(X(I).LT.XMIN) XMIN=X(I)
C3170       IF(X(I).GT.XMAX) XMAX=X(I)
C3180       IF(Y(I).LT.YMIN) YMIN=Y(I)
C3190       IF(Y(I).GT.YMAX) YMAX=Y(I)
C3200  6    CONTINUE
C3210       DEL=XMAX-XMIN
C3220       IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C3230       XMAX=XMIN+DEL
C3240       YMAX=YMIN+DEL
C3250       DO 5 I =1,N10
C3260       Z=I
C3270     5 HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
C3280       DY=(YMAX-YMIN)/(NR-1)
C3310       Z=YMAX+DY
C3320       YL=Z-DY/2.
```

```
C3325         IBLK=0
C3330         DO 7 J=1,NR
C3340         DO 8 K=1,NC
C3350  8      LINE(K)=BLANK
C3360         Z=Z-DY
C3370         YU=YL
C3380         YL=Z-DY/2.
C3390         DO 9 I=1,N
C3400         IF(Y(I).GE.YU) GO TO 9
C3410         IF(Y(I).LT.YL) GO TO 9
C3415         IBLK=1
C3420         K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
C3430         IF(K.GT.NC) K=NC
C3440         LINE(K)=STAR
C3480  9      CONTINUE
C3490         IF(IBLK.NE.0) WRITE(108,508) Z,(LINE(K),K=1,NC)
C3500  7      CONTINUE
C3510         WRITE(108,504)
C3520         WRITE( 108,3002)
C3530         WRITE( 108,507) XMIN,(HEAD(I),I=1,N10)
C3540         RETURN
C3550   504  FORMAT ( 1X, 14(1H-), 1H., 10(5H----.), 1H- )
C3560   507  FORMAT(10X,11(F10.4))
C3570   508  FORMAT (1X, F12.4,1X, 1HI, 51A1, 1HI )
C3580  3002  FORMAT(4X,7H R / Z ,4X,1HI,5(9X,1HI))
C3590         END
```



```
C0100BOTZ,P30,T100,CM140000,STCX3.
C0110ACCOUNT,M0326S,BOT24.
C0120BANNERS(OUTPUT) *J. OUTNAM*DEPT 220*BLD 110-4**
C0130DEFINE(OUTFIL=ZSSX)
C0138FTN(R=0,OPT=2)
C0140LDSET(PRESET=INDEF)
C0150LGO.
C0210EXIT.
C0230/EOR
C0240       PROGRAM BOTZSS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C0250      1 OUTFIL,TAPE7=OUTFIL)
C0252C
C0254C BOTZSS GENERATES THE BOT Z MATRIX.
C0256C
C0258C UNIT 5 IS THE CARD READER.
C0260C UNIT 6 IS THE LINE PRINTER.
C0262C UNIT 7 IS A DISK FILE FOR OUTPUT OF THE Z MATRIX.
C0264C
C0280       COMPLEX A3,Z(6400),G(3403),GB(3403),U
C0281       COMPLEX GS,GBS
C0282       DIMENSION TWGHT(40),ZWGHT(40)
C0283       COMMON /WAVE/ BK
C0284       COMMON /BOT1/ NMODE,NPT,NBAND
C0286       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C0288       COMMON /BOT3/ DH(82),SV(82),CV(82)
C0290       COMMON /BOT5/ T(160),TP(160),TZ(160)
C0291       COMMON /BOT6/ IEDGE,IUNIF
C0320       COMMON /INT/ M,N,BKL,RHO2,DELTA
C0325       COMMON /UCONST/ AMN,ABMN,PIM,PIN
C0326       DIMENSION LI(2),LJ(2),NAMI(2),NAMJ(2)
C0327       EXTERNAL FUNC,FUNCB,FUNCS,FUNCBS
C0328       DATA NAMI,NAMJ /4HS(T),4HS(Z),4HS(T),4HS(Z)/
C0430       U=(0.,1.)
C0432       PI=3.14159265
C0434       ETA=376.707
C0435       READ(5,1) BK
C0436 1     FORMAT(E15.7)
C0437       WRITE(6,2) BK
C0438 2     FORMAT(9H1       BK,//,E15.7)
C0900C
C0510       CALL BOTIN
C0520C
C0530       WRITE(7) NP
C0790       KG=NP-1
C0810       NM=(NP-3)/2
C0820       NM2=NM*2
C0832       LSS=NM2*NM2
C0833       NG=KG*KG
C0834       BKL=BK*BL
C0835       BK2=BK*BK
C0836C CONSTANTS USED IN SUMMATION OF IMPEDANCE ELEMENTS.
C0840       C1=BK*ETA
```

```
      C2=PI*ETA/BK/BL
      C3=C2
      C4=C1
      B4=(PI/BK/BL)**2
C***  *** END INITIALIZATION OF PARAMETERS ***
C     CALCULATE IMPEDANCE LOADING WEIGHTS ON BOT.
      DO 10 J=1,NM
      J2=2*(J-1)+1
      J3=J2+1
      J4=J3+1
      J5=J4+1
      J6=4*(J-1)+1
      J7=J6+1
      J8=J7+1
      J9=J8+1
      TWGHT(J)=2.*BL*(DH(J2)*T(J6)+DH(J3)*T(J7)+
     1 DH(J4)*T(J8)+DH(J5)*T(J9))
      ZWGHT(J)=2.*BL*(DH(J2)*TZ(J6)+DH(J3)*TZ(J7)+
     1 DH(J4)*TZ(J8)+DH(J5)*TZ(J9))
10    CONTINUE
      WRITE(7) (TWGHT(J),J=1,NM)
      WRITE(7) (ZWGHT(J),J=1,NM)
C***  COMPUTATION OF IMPEDANCE MATRICES Z(M,N), WHERE M = 0 TO NMODE-1
C     AND N = -M TO +M.
C***  NOTE, THESE MATRICES ARE COMPUTED IN THE FOLLOWING ORDER:
C     1) ALONG THE MAIN DIAGONAL (M=N)
C     2) ALONG THE FIRST OFF-DIAGONAL (M-N=1)
C     3) ALONG THE SECOND OFF-DIAGONAL (M-N=2)
C     ETC.
      IMODE=NMODE
      KMODE=2*NMODE-1
C***  THIS LOOP FIXES ONE OF THE DIAGONALS.
      DO 999 NN=1,KMODE
      IF(NN.GT.NBAND) GO TO 999
      IF(MOD(NN,2).EQ.0) GO TO 20
      M=(NN+1)/2-2
      N=-(NN+1)/2
      GO TO 21
20    IMODE=IMODE-1
      M=NN/2-1
      N=-NN/2
21    CONTINUE
C***  THIS LOOP STEPS M AND N ALONG THE FIXED DIAGONAL.
      DO 998 MM=1,IMODE
      M=M+1
      N=N+1
      WRITE(6,54) M,N
54    FORMAT(4H1 M=,I3,4H  N=,I3)
C***  CONSTANTS USED IN AMN FUNCTIONS.
      IF(M.EQ.N) AMN=0.5/PI
      IF(M.NE.N) AMN=(-1)**(M-N+1)/FLOAT(N-M)/PI/PI/2.
      AMMN=(-1)**(M-N)/PI/2.
```

```
C1020       PIM=PI*M
C1030       PIN=PI*N
C1185C COMPUTATION OF GREEN'S FUNCTION KERNEL FOR J.GE.I.
C1186C NOTE, G IS A SYMMETRIC MATRIX, HENCE ONLY THE UPPER TRIANGULAR
C1187C   PORTION IS STORED BY COLUMNS FROM G(1) TO G(KG*(KG+1)/2).
C1188C   G(I,J) IS STORED IN G(I+(J-1)*J/2) WHEN I.LE.J.
C1190       I1=0
C1192       NPT=0
C1194       NINT=0
C1196       NPTS=0
C1198       NINTS=0
C1228       DO 16 J=1,KG
C1230       DO 17 I=1,J
C1240       I1=I1+1
C1260       YY=YB1(J)-YB1(I)
C1270       XX=XB1(J)-XB1(I)
C1280       X1=ABS(XX)+ABS(YY)
C1285       IF(X1.NE.0.0) GO TO 7
C1288C SELF TERM INTEGRATION.
C1290       IF(J.NE.1 .AND. IUNIF.EQ.1) GO TO 6
C1291       DELTA=DH(I)/2./BL
C1292       CALL CSIMP(FUNCS,0.0,2.0,0.05,10,A3,G(I1),K,IER)
C1293       NPTS=NPTS+K+1
C1294       NINTS=NINTS+1
C1295       G(I1)=G(I1)*2.*BL*BL/DH(I)
C1296       GS=G(I1)
C1298       IF(N.EQ.0) GO TO 17
C1299       CALL CSIMP(FUNCBS,0.0,2.0,0.05,10,A3,GB(I1),K,IER)
C1300       NPTS=NPTS+K+1
C1301       NINTS=NINTS+1
C1302       GB(I1)=GB(I1)*2.*BL*BL/DH(I)
C1303       GBS=GB(I1)
C1304       GO TO 17
C1305C BOT SEGMENTATION IS UNIFORM(IUNIF=1).
C1306  6    G(I1)=GS
C1307       IF(N.NE.0) GB(I1)=GBS
C1308       GO TO 17
C1309C NON-SELF TERM INTEGRATION.
C1310  7    RHO2=(XX*XX+YY*YY)/BL/BL
C1320       CALL CSIMP(FUNC,0.0,2.0,0.05,10,A3,G(I1),K,IER)
C1322       NPT=NPT+K+1
C1324       NINT=NINT+1
C1330       G(I1)=G(I1)*BL
C1340       IF(N.EQ.0) GO TO 17
C1350       CALL CSIMP(FUNCB,0.0,2.0,0.05,10,A3,GB(I1),K,IER)
C1352       NPT=NPT+K+1
C1354       NINT=NINT+1
C1360       GB(I1)=GB(I1)*BL
C1650   17  CONTINUE
C1660   16  CONTINUE
C1662       AVG=NPT/FLOAT(NINT)
C1664       AVGS=NPTS/FLOAT(NINTS)
```

```
C1666       WRITE(6,56) MINTS,AVGS,NINT,AVG
C1668  56   FORMAT(/,' NUMBER OF INTEGRATIONS AND AVERAGE NUMBER OF',
C1670      1 ' POINTS PER INTEGRATION (SELF AND NON-SELF TERMS,',
C1672      2 ' RESPECTIVELY)',/,2(I8,G10.3))
C1674C *** END CALCULATION OF GREENS'S KERNEL ***
C2005C COMPUTATION OF IMPEDANCE ELEMENTS, EQS. 17-20.
C2006C NOTE, THE SYMMETRIES IN THE SUBMATRICES OF Z(P,N) ARE USED.
C2007C THESE SYMMETRIES ARE: ZTT = TRANSPOSE(ZTT),
C2008C ZZZ = TRANSPOSE(ZZZ), AND M * TRANSPOSE(ZTZ) = -N * ZZT.
C2010       DO 30 J=1,NM
C2020       JL=(J-1)*NM2
C2030       J3=(J-1)*4
C2040       J1=2*(J-1)
C2050       DO 31 I=1,J
C2055C COMPUTE SUBMATRIX INDICES.
C2060       L1=JL+I
C2070       L2=L1+NM
C2080       L3=NM*NM2+L1
C2090       L4=L3+NM
C2100       Z(L1)=0.
C2110       Z(L2)=0.
C2120       Z(L3)=0.
C2130       Z(L4)=0.
C2135       IF(N.EQ.0 .AND. M.EQ.0 .AND. I.EQ.J) Z(L4)=1.0
C2140       IF(IABS(I-J).GE.NPT) GO TO 28
C2150       I1=2*(I-1)
C2160       I3=(I-1)*4
C2170       DO 70 JJ=1,4
C2180       J2=J1+JJ
C2190       J4=J3+JJ
C2200       DO 71 II=1,4
C2210       I2=I1+II
C2220       I4=I3+II
C2224C THE COMPUTATION OF J4 REFLECTS THE FACT THAT G IS SYMMETRIC,
C2225C   WITH ONLY THE UPPER PORTION STORED AS NOTED ABOVE.
C2226C NOTE, IF G WAS FILLED COMPLETELY, THEN J4=(J2-1)*KG+I2.
C2230       J4=MAXC(I2,J2)
C2240       J4=(J4-1)*J472+MINC(I2,J2)
C2260       SS=SV(I2)*SV(J2)
C2270       CC=CV(I2)*CV(J2)
C2295C IMPEDANCE ELEMENTS COMPUTED, EQS. 17-20.
C2296C Z(L1) = ZTT, Z(L2) = ZTT, Z(L3) = ZTZ, Z(L4) = ZZZ.
C2300       Z(L1)=Z(L1)+DH(I2)*DH(J2)*(T(I7)*T(J7)*(CC+SS)-TP(I7)*
C2305      1  TP(J7)/RK2)*G(J4)
C2310       Z(L2)=Z(L2)+DH(I2)*DH(J2)*TZ(I7)*TP(J7)*G(J4)
C2315       Z(L3)=Z(L3)+DH(I2)*DH(J2)*TP(I7)*TZ(J7)*G(J4)
C2325       IF(N.EQ.0) GO TO 71
C2330       Z(L4)=Z(L4)+DH(I2)*DH(J2)*TZ(I7)*TZ(J7)*
C2335      1  ((1.-R*N*B4)*G(J4)+GB(J4))
C2370    71 CONTINUE
C2375    70 CONTINUE
C2379C MULTIPLY BY CONSTANTS, EXCEPT FOR N AND M FACTORS IN ZTZ, AND
```

```
02380C ZZT, RESPECTIVELY.
02381        Z(L1)=U*C1*Z(L1)
02382        Z(L2)=C2*Z(L2)
02383        Z(L3)=C3*Z(L3)
02384        IF(N.NE.0) Z(L4)=U*C4*Z(L4)
02385  28    CONTINUE
02387C FILL THE Z(M,N) SUBMATRIX USING THE ZTT, ZTZ, ZZT, AND
02388C ZZZ SYMMETRIES NOTED ABOVE.
02390C       IF(I.EQ.J) GO TO 29
02392C COMPUTE SUBMATRIX TRANSPOSE INDICES.
02395        L1T=(I-1)*NM2+J
02396        L2T=L1T+NM
02397        L3T=NM*NM2+L1T
02398        L4T=L3T+NM
02399        Z(L1T)=Z(L1)
02400        Z(L2T)=-M*Z(L3)
02401        Z(L3T)=N*Z(L2)
02402        Z(L4T)=Z(L4)
02404  29    Z(L2)=-M*Z(L2)
02405        Z(L3)=N*Z(L3)
02406  31    CONTINUE
02407  30    CONTINUE
02408C *** END CALCULATION OF Z(M,N) SUBMATRIX ***
02409C PRINT OUT IMPEDANCE MATRIX Z(M,N) BY SUBMATRIX BY COLUMNS.
02500        LI(1)=NM
02502        LI(2)=NM
02504        LJ(1)=NM
02506        LJ(2)=NM
02508        CALL ZLIST(2,2,LI,LJ,NAMI,NAMJ,Z)
02550C WRITE Z(M,N) TO DISK.
02560        WRITE(7) M,N
02570        WRITE(7) (Z(I),I=1,LSS)
02590  998   CONTINUE
02730  999   CONTINUE
02740        STOP
02750        END
02760C ------------------------------------------------------------
02770        SUBROUTINE BOTIN
02780C
02790C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
02800C
02810        COMMON /BOT1/ NMODE,NPT,NBAND
02820        COMMON /BOT2/ NP,RL,YB(83),XB(83),YB1(82),XB1(82)
02830        COMMON /BOT3/ DH(82),SV(82),CV(82)
02850        COMMON /BOT5/ T(160),TP(160),TZ(160)
02860        COMMON /BOT6/ IEDGE,IUNIF
02870        READ(5,49) NMODE,NPT,NBAND
02880        READ(5,49) NP
02890  49    FORMAT(3I3)
02900        WRITE(6,48) NMODE,NPT,NBAND,NP
02910  48    FORMAT(32H     NMODE    NPT     NBAND     NP,/,4I8)
02940        READ(5,53)(YB(I),I=1,NP)
```

```
      READ(5,53)(XB(I),I=1,NP)
   53 FORMAT(10F8.4)
      WRITE(6,55)
   55 FORMAT(/,3H YB)
      WRITE(6,46)(YB(I),I=1,NP)
   46 FORMAT(1X,10F8.4)
      WRITE(6,56)
   56 FORMAT(/,3H XB)
      WRITE(6,46)(XB(I),I=1,NP)
C PLOT THE BODY COORDINATES.
      CALL PLOTB(XB,YB,NP,41)
      READ(5,53) BL
      WRITE(6,47) BL
   47 FORMAT(//,21H HALF-LENGTH OF BOT =,F12.4)
C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
      IF(MOD(NP,2).NE.1) GO TO 980
C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
      IEDGE=1
      IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
      IEDGE=0
      YB(NP+1)=YB(2)
      XB(NP+1)=XB(2)
      YB(NP+2)=YB(3)
      XB(NP+2)=XB(3)
      NP=NP+2
      WRITE(6,66) NP
   66 FORMAT(//,39H BOT GENERATING CURVE IS CLOSED.  NP = ,I3)
   10 CONTINUE
C COMPUTATION OF BODY SEGMENT PARAMETERS.
      DO 57 I=2,NP
      I2=I-1
      RR1=YB(I)-YB(I2)
      RR2=XB(I)-XB(I2)
      DH(I2)=SQRT(RR1*RR1+RR2*RR2)
      XB1(I2)=.5*(XB(I)+XB(I2))
      YB1(I2)=.5*(YB(I)+YB(I2))
      SV(I2)=RR1/DH(I2)
      CV(I2)=RR2/DH(I2)
   57 CONTINUE
C CHECK IF BOT SEGMENTATION IS UNIFORM.
      IUNIF=0
      NP1=NP-1
      DO 60 I=2,NP1
      RR1=DH(I)/DH(1)
      IF(RR1.LT.0.99 .OR. RR1.GT.1.01) GO TO 20
   60 CONTINUE
      IUNIF=1
      WRITE(6,67)
   67 FORMAT(//,' BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
   20 CONTINUE
C COMPUTATION OF TRIANGLE FUNCTIONS T.
      NM=(NP-3)/2
```

```
C3500       DO 74 J=1,NM
C3510       J2=2*(J-1)+1
C3520       J3=J2+1
C3530       J4=J3+1
C3540       J5=J4+1
C3550       J6=4*(J-1)+1
C3560       J7=J6+1
C3570       J8=J7+1
C3580       J9=J8+1
C3590       DEL1=DH(J2)+DH(J3)
C3600       DEL2=DH(J4)+DH(J5)
C3610       TP(J6)=1./DEL1
C3620       TP(J7)=1./DEL1
C3630       TP(J8)=-1./DEL2
C3640       TP(J9)=-1./DEL2
C3650       T(J6)=DH(J2)/2./DEL1
C3660       T(J7)=(DH(J2)+DH(J3)/2.)/DEL1
C3670       T(J8)=(DH(J4)/2.+DH(J5))/DEL2
C3680       T(J9)=DH(J5)/2./DEL2
C3690    74 CONTINUE
C3700       NM4=NM*4
C3710       DO 75 J=1,NM4
C3720    75    TZ(J)=T(J)
C3730       IF(IEDGE.EQ.0) GO TO 76
C3740       TZ(1)=2.-T(1)
C3750       TZ(2)=2.-T(2)
C3760       TZ(NM4-1)=2.-T(NM4-1)
C3770       TZ(NM4)=2.-T(NM4)
C3780    76    CONTINUE
C3790       RETURN
C3800   980 WRITE(6,981)
C3810   981 FORMAT(//,' **** ERROR IN BOT INPUT')
C3820       STOP
C3830       END
C3840C ***************************************************************
C3850       SUBROUTINE CSIMP(F,A,B,DEL,IMAX,SI1,S,N,IER)
C3860C ***************************************************************
C3870C COMPLEX SIMPSON INTEGRATION ROUTINE.
C3880       COMPLEX F,SI1,S,SUMK
C3890       SI1=(1.0,.0)
C3900       S=(1.C,.0)
C3910       N=0
C3920       BA=B-A
C3930       IF (BA)20,19,20
C3940    19 IER=1
C3950       RETURN
C3960    20 IF(DEL)22,22,23
C3970    22 IER=2
C3980       RETURN
C3990    23 IF(IMAX-1)24,24,25
C4000    24 IER=3
C4010       RETURN
```

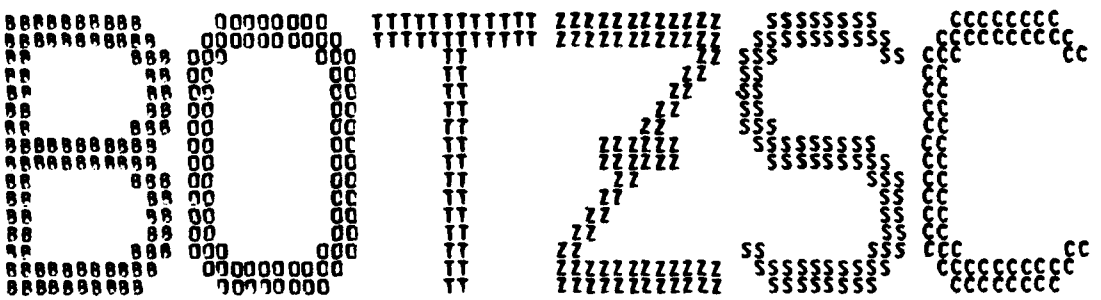
```
C4020     25 X=BA/2.+A
C4030        NHALF=1
C4040        SUMK=F(X)*BA*2./3.
C4050        S=SUMK+(F(A)+F(B))*BA/6.
C4060        DO 28 I=2,IMAX
C4070        SI1=S
C4080        S=(S-SUMK/2.)/2.
C4090        NHALF=NHALF*2
C4100        ANHLF=NHALF
C4110        FRSTX=A+(BA/ANHLF)/2.
C4120        SUMK=F(FRSTX)
C4130        XK=FRSTX
C4140        KLAST=NHALF-1
C4150        FINC=BA/ANHLF
C4160        DO 26 K=1,KLAST
C4170        XK=XK+FINC
C4180     26 SUMK=SUMK+F(XK)
C4190        SUMK=SUMK*2.*BA/(3.*ANHLF)
C4200        S=S+SUMK
C4205        IF(I.EQ.2) GO TO 28
C4210        IF(CABS(S).EQ.0.0) GO TO 29
C4220        IF((CABS(S-SI1)/CABS(S))-DEL) 29,28,28
C4230     28 CONTINUE
C4240        IER=4
C4250        WRITE(6,1) SI1,S
C4260  1     FORMAT(///,42H INTEGRATION OF SELF TERM DID NOT CONVERGE,//
C4270       1 50H THE PREVIOUS AND FINAL VALUES OF THE INTEGRAL ARE,
C4280       2 13H RESPECTIVELY,4E12.3)
C4290        GO TO 30
C4300     29 IER=0
C4310     30 N=2*NHALF
C4320        RETURN
C4330        END
C4340C ------------------------------------------------
C4350        COMPLEX FUNCTION FUNC(E)
C4360C G(NON-SELF TERM) INTEGRAND.
C4370        COMPLEX U
C4380        COMMON /INT/ M,N,BKL,RHO2,DELTA
C4390        DATA U /(0.,1.)/
C4400        R=SQRT(RHO2+E*E)
C4410        FUNC=CEXP(-U*BKL*R)/R*UMN(M,N,E)
C4420        RETURN
C4430        END
C4440C ------------------------------------------------
C4450        COMPLEX FUNCTION FUNCB(E)
C4460C GB(NON-SELF TERM) INTEGRAND.
C4470        COMPLEX U
C4480        COMMON /INT/ M,N,BKL,RHO2,DELTA
C4490        DATA U /(0.,1.)/
C4500        R=SQRT(RHO2+E*E)
C4510        FUNCB=CEXP(-U*BKL*R)/R*UBMN(M,N,E)
C4520        RETURN
```

```
      END
C-----------------------------------------------
      FUNCTION UMN(M,N,E)
      COMMON /UCONST/AMN,ABMN,PIM,PIN
      IF(M.NE.N) GO TO 100
      UMN=AMN*(2.-E)*COS(PIN*E)
      RETURN
  100 UMN=AMN*(SIN(PIN*E)-SIN(PIM*E))
      RETURN
      END
C-----------------------------------------------
      FUNCTION UMNP(M,N,E)
      COMMON /UCONST/AMN,ABMN,PIM,PIN
      IF(M.NE.N) GO TO 100
      UMNP=AMN*(-PIN*(2.-E)*SIN(PIN*E)-COS(PIN*E))
      RETURN
  100 UMNP=AMN*(PIN*COS(PIN*E)-PIM*COS(PIM*E))
      RETURN
      END
C*************************************************************
      COMPLEX FUNCTION FUNCS(E)
C*************************************************************
C G(SELF TERM) INTEGRAND.
      COMPLEX U
      COMMON /INT/ M,N,BKL,RHO2,DELTA
      DATA PI,U /3.14159265,(0.,1.)/
      IF(E.EQ.0.0) GO TO 100
      AE=ALOG(E)
      A=ALOG(DELTA+SQRT(DELTA*DELTA+E*E))
      FUNCS=UMN(M,N,E)*(-U*BKL*DELTA+A+U*BKL*E*(A-2.*AE+1.))
      FUNCS=FUNCS+UMNP(M,N,E)*E*(AE-1.)
      FUNCS=FUNCS*CEXP(-U*BKL*E)
      RETURN
  100 CONTINUE
      FUNCS=UMN(M,N,0.0)*(-U*BKL*DELTA+ALOG(2.*DELTA))
      RETURN
      END
C*************************************************************
      COMPLEX FUNCTION FUNCBS(E)
C*************************************************************
C GB(SELF TERM) INTEGRAND.
      COMPLEX U
      COMMON /INT/ M,N,BKL,RHO2,DELTA
      DATA PI,U /3.14159265,(0.,1.)/
      IF(E.EQ.0.0) GO TO 100
      AE=ALOG(E)
      A=ALOG(DELTA+SQRT(DELTA*DELTA+E*E))
      FUNCBS=UBMN(M,N,E)*(-U*BKL*DELTA+A+U*BKL*E*(A-2.*AE+1.))
      FUNCBS=FUNCBS+UBMNP(M,N,E)*E*(AE-1.)
      FUNCBS=FUNCBS*CEXP(-U*BKL*E)
      RETURN
  100 CONTINUE
```

```
C5050       FUNCBS=U8MN(M,N,0.C)*(-U*RKL*DELTA+ALOG(2.*DELTA))
C5060       RETURN
C5070       END
C5080C ------------------------------------------
C5090       FUNCTION U8MN(M,N,E)
C5100C NOTE, M.NE.C  AND  N.NE.0
C5110       COMMCN /UCONST/AMN,ABMN,PIM,PIN
C5120       U8MN=ABMN*(2.-E+SIN(PIN*E)/PIN+SIN(PIM*E)/PIM)
C5130       RETURN
C5140       END
C5150C ------------------------------------------------
C5160       FUNCTION U8MNP(M,N,E)
C5170C NOTE, M.NE.C  AND  N.NE.0
C5180       COMMCN /UCONST/AMN,ABMN,PIM,PIN
C5190       U8MNP=ABMN*(-1.+COS(PIN*E)+COS(PIM*E))
C5200       RETURN
C5210       END
C5220C ********************************************************
C5230       SUBROUTINE PLOTB(X,Y,N,NR)
C5240C ********************************************************
C5250C
C5260C WRITTEN 2/14/74   BY J. M. PUTNAM       DEPT 220     X23877
C5270C
C5280C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C5290C    N IS THE NUMBER OF POINTS TO BE PLOTTED.
C5300C    NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C5310C    NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C5320C       NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C5330C
C5340       REAL X(1),Y(1),HEAD(10)
C5350       INTEGER  LINE(101),BLANK,STAR,PLUS
C5360       DATA BLANK,STAR,PLUS /1H ,1H*,1H+/
C5370       NC=51
C5380       N10=(NC-1)/10
C5390       WRITE(6,500)
C5400 500   FORMAT(///,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
C5410       WRITE(6,504)
C5420       XMIN=X(1)
C5430       XMAX=X(1)
C5440       YMIN=Y(1)
C5450       YMAX=Y(1)
C5460       DO 6 I=1,N
C5470       IF(X(I).LT.XMIN) XMIN=X(I)
C5480       IF(X(I).GT.XMAX) XMAX=X(I)
C5490       IF(Y(I).LT.YMIN) YMIN=Y(I)
C5500       IF(Y(I).GT.YMAX) YMAX=Y(I)
C5510 6     CONTINUE
C5520       DEL=XMAX-XMIN
C5530       IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C5540       XMAX=XMIN+DEL
C5550       YMAX=YMIN+DEL
C5560       DO 5 I =1,N10
```

```
C5570         Z=I
C5580       5 HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
C5590         DY=(YMAX-YMIN)/(NR-1)
C5600         IEDGE=1
C5610         IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
C5620         Z=YMAX+DY
C5630         YL=Z-DY/2.
C5640         DO 7 J=1,NR
C5650         DO 8 K=1,NC
C5660   8     LINE(K)=BLANK
C5670         Z=Z-DY
C5680         YU=YL
C5690         YL=Z-DY/2.
C5700         DO 9 I=1,N
C5710         IF(Y(I).GE.YU) GO TO 9
C5720         IF(Y(I).LT.YL) GO TO 9
C5730         K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
C5740         IF(K.GT.NC) K=NC
C5750         LINE(K)=STAR
C5760         IF(MOD(I,2).EQ.1) LINE(K)=PLUS
C5770         IF(IEDGE.EQ.0) GO TO 9
C5780         IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
C5790   9     CONTINUE
C5800         WRITE(6,508) Z,(LINE(K),K=1,NC)
C5810   7     CONTINUE
C5820         WRITE(6,504)
C5830         WRITE( 6,3002)
C5840         WRITE( 6,507) XMIN,(HEAD(I),I=1,N10)
C585C         RETURN
C5860   504 FORMAT ( 1X, 14(1H-), 1H., 10(5H----.), 1H- )
C5870   507 FORMAT(10X,11(F10.4))
C588C   508 FORMAT (1X, F12.4,1X, 1HI, 51A1, 1HI )
C5890  3002 FORMAT(4X,7HYH / XH,4X,1HI,5(9X,1HI))
C5900         END
C5910C ---------------------------------------------
C5920         SUBROUTINE ZLIST(NI,NJ,LI,LJ,NAMI,NAMJ,Z)
C5930C WRITE Z TO THE LINE-PRINTER BY SUBMATRIX BY COLUMNS.
C5940         COMPLEX Z(1)
C5950         DIMENSION LI(1),LJ(1),NAMI(1),NAMJ(1)
C5960         NCOL=0
C5970         DO 100 I=1,NI
C5980  100    NCOL=NCOL+LI(I)
C5990         NR=1
C6000         DO 300 J=1,NJ
C6010         IF(LJ(J).EQ.0) GO TO 300
C6020         NC=1
C6030         DO 200 I=1,NI
C6040         IF(LI(I).EQ.0) GO TO 200
C6050         WRITE(6,1) NAMI(I),NAMJ(J)
C6060   1     FORMAT(/,2H Z,1X,A4,3H - ,A4)
C6070         KI=LI(I)
C6080         KJ=LJ(J)
```

```
C6090         K1=(NR-1)*NCOL+NC
C6100         DO 15C KK=1,KJ
C6110         K2=K1+KI-1
C6120         WRITE(6,2) (Z(K),K=K1,K2)
C6130 2       FORMAT(1X,10G11.4)
C6140 15C     K1=K1+NCOL
C6150         NC=NC+LI(I)
C6160 2CC     CONTINUE
C6170         NR=NR+LJ(J)
C6180 30C     CONTINUE
C6190         RETURN
C6200         END
C6210/EOR
```



```
00100BOTZ,P30,T100,CM146000,STCX3.
00110ACCOUNT,HO3265,BOT24.
00120BANNERS(OUTPUT) *J. PUTNAM*DEPT 220*BLD 110-4**
00130DEFINE(OUTFIL=ZSCX)
00138FTN(R=0,OPT=2)
00140LDSET(PRESET=INDEF)
00150LGO.
00210EXIT.
00230/EOR
00240       PROGRAM BOTZSC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
00250      1 OUTFIL,TAPE7=OUTFIL)
00252C
00254C  BOTZSC GENERATES THE BOT-CAP Z MATRIX.
00256C
00258C  UNIT 5 IS THE CARD READER.
00260C  UNIT 6 IS THE LINE PRINTER.
00262C  UNIT 7 IS A DISK FILE FOR OUTPUT OF THE Z MATRIX.
00264C
00280       COMPLEX U,A3,Z(1000),G2(3000)
00281       COMPLEX Z1,Z2,Z3,Z4
00282       COMPLEX Z3E,Z4E
00283       COMPLEX G1E(1500),G2E(1500)
00284       COMMON /WAVE/ BK
00285       COMMON /BOT1/ NMODE,NPT,NBAND
00287       COMMON /BOT2/ NP,BL,YR(83),XR(83),YB1(82),XB1(82)
00288       COMMON /BOT3/ DH(82),SV(82),CV(82)
00289       COMMON /BOT5/ T(160),TP(160),TZ(160)
00296       COMMON /CAP1/ NC,XC,YC,ZC(2)
00292       COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
00294       COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
00296       COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
00298       COMMON /EDG1/ NE,ZE(2),ZRE(10)
00300       COMMON /EDG2/ TCE(10),TPCE(10),TRE(10),TPBE(10)
00320       COMMON /INT/ M,RHO2,ZP
00325       DIMENSION LI(2),LJ(3),NAMI(2),NAMJ(3)
00330       EXTERNAL FUNC1,FUNC2
00340       DATA NAMI,NAMJ /4HS(I,4HS(Z),4HC(I),4HC(R),4HEDGE/
00436       U=(0.,1.)
00432       PI=3.14159265
00434       ETA=376.707
00435       READ(5,1) BK
00436  1    FORMAT(E15.7)
00437       WRITE(6,2) BK
00438  2    FORMAT(9H1       BK,/,E15.7)
00440C
00450       CALL BOTIN
00460C
00470       CALL CAPIN
00480C
00485       WRITE(7) NP,NC,NPR,NE
00490       NM=(NP-3)/2
00495       NM2=NM*2
```

```
C0500C       LR=(NPR-3)/2
C0510        LC=NC*NM*LR
C0520        LC2=LC*2
C0525        LE=NE*NM
C0530        LSC=NM2*(LC2+LE)
C0540        NP1=NP-1
CC550        NPR1=NPR-1
C0551        BKL=BK*BL
C0552        BK2=BK*BK
C0553C CONSTANTS USED IN SUMMATION OF IMPEDANCE ELEMENTS.
C0554        C1=BK*ETA
C0555        C2=PI*ETA/BK/BL
C0556        C3=C1
C0557        C4=C2
C0558        C3E=ETA/BK
C0559        C4E=C1
C0560C *** END INITIALIZATION OF PARAMETERS ***
C0562C COMPUTATION OF IMPEDANCE MATRICES Z(M), WHERE M = -NMODE+1 TO
C0564C NMODE-1.
C0566        M=-NMODE
C0568        KMODE=2*NMODE-1
C0570        DO 998 MM=1,KMODE
C0572        M=M+1
C0574        WRITE(6,54) M
C0576 54     FORMAT(4H1 M=,I3)
C0578C COMPUTATION OF GREEN'S FUNCTION KERNEL.
C0600        I1=0
C0605        NPT2=0
C0610        NINT2=0
C0615        DO 16 JC=1,NC
C0630        DO 16 JP=1,NP1
C0640        DO 16 JR=1,NPR1
C0645        DO 17 IP=1,NP1
C0650        I1=I1+1
C0660        YY=YB1(IP)-(YC+RHOC1(JR)*RC1(JP)*SPC(JP))
C0670        XX=XB1(IP)-(XC+RHOC1(JR)*RC1(JP)*CPC(JP))
C0680        RHO2=XX*XX+YY*YY
C0690        ZP=ZC(JC)
C0700        CALL CSIMP(FUNC2,-BL,BL,0.05,10,A3,G2(I1),K,IER)
C0710        NPT2=NPT2+K+1
C0720        NINT2=NINT2+1
C0730        G2(I1)=G2(I1)/4./PI
C0740 17     CONTINUE
C0750 16     CONTINUE
C0760        AVG2=NPT2/FLOAT(NINT2)
C0765        WRITE(6,56) NINT2,AVG2
C0770 56     FORMAT(/,' NUMBER OF INTEGRATIONS AND AVERAGE NUMBER OF',
C0775       1' POINTS PER INTEGRATION',/,(I8,G10.3))
C0777        IF(NE.EQ.0) GO TO 60
C0780        I1=0
C0785        NPT1=0
C0790        NINT1=0
```

```
C0795       NPT2=0
C0800       NINT2=0
C0805       DO 18 JC=1,NC
C0810       DO 18 JP=1,NP1
C0815       DO 18 JZ=1,2
C0820       DO 19 IP=1,NP1
C0825       I1=I1+1
C0830       YY=YB1(IP)-YB1(JP)
C0835       XX=XB1(IP)-XB1(JP)
C0840       RHO2=XX*XX+YY*YY
C0845C CHECK FOR SELF-TERM.
C0850       IF(RHO2.EQ.0.0) RHO2=DH(IP)*DH(IP)/4.
C0855       ZP=ZBE((JC-1)*2+JZ)
C0860       CALL CSIMP(FUNC2,-BL,BL,0.05,10,A3,G2E(I1),K,IER)
C0865       NPT2=NPT2+K+1
C0870       NINT2=NINT2+1
C0875       G2E(I1)=G2E(I1)/4./PI
C0880       IF(MM.NE.1) GO TO 19
C0885       CALL CSIMP(FUNC1,-BL,BL,0.05,10,A3,G1E(I1),K,IER)
C0890       NPT1=NPT1+K+1
C0895       NINT1=NINT1+1
C0900       G1E(I1)=G1E(I1)/4./PI
C0905 19    CONTINUE
C0910 18    CONTINUE
C0915       AVG1=0.0
C0920       IF(MM.EQ.1) AVG1=NPT1/FLOAT(NINT1)
C0925       AVG2=NPT2/FLOAT(NINT2)
C0930       WRITE(6,57) NINT1,AVG1,NINT2,AVG2
C0935 57    FORMAT(/,' NUMBER OF INTEGRATIONS AND AVERAGE NUMBER OF ',
C0940      1 ' POINTS PER INTEGRATION (G1 AND G2, RESPECTIVELY)',/,
C0945      2 2(I8,G10.3))
C0947 60    CONTINUE
C0950C *** END CALCULATION OF GREEN'S KERNEL ***
C0951C BOT-CAP LOOP.
C0952       DO 30 JC=1,NC
C0954       DO 30 JP=1,NM
C0956       DO 30 JR=1,LR
C0958       L1=((JC-1)*NM*LR+(JP-1)*LR+(JR-1))*NM2
C0960       L2=L1+NM
C0962       L3=L1+NM2*LC
C0964       L4=L3+NM
C0968       DO 31 IP=1,NM
C0972       L1=L1+1
C0974       L2=L2+1
C0976       L3=L3+1
C0978       L4=L4+1
C0980       Z1=0.0
C0982       Z2=0.0
C0984       Z3=0.0
C0986       Z4=0.0
C0990       JP2=2*(JP-1)
C0992       JP4=4*(JP-1)
```

```
00994       DO 70 JJP=1,4
00996       JP2=JP2+1
00998       JP7=JP7+1
01000       JR2=2*(JR-1)
01002       JR7=4*(JR-1)
01004       DO 70 JJR=1,4
01006       JR2=JR2+1
01008       JR7=JR7+1
01010       IP2=2*(IP-1)
01012       IP7=4*(IP-1)
01014       DO 71 IIP=1,4
01016       IP2=IP2+1
01018       IP7=IP7+1
01020       AJ=AC(JP2)*ABS(RHOC(JR2+1)**2-RHOC(JR2)**2)
01025       RJ=RHOC1(JR2)*RC1(JP2)
01030       J4=((JC-1)*NP1*NPR1+(JP2-1)*NPR1+(JR2-1))*NP1+IP2
01192C BOT-CAP IMPEDANCE ELEMENTS COMPUTED.
01202C Z(L1) = ZTT , Z(L2) = ZZT , Z(L3) = ZTR , Z(L4) = ZZR.
01210       Z1=Z1+DH(IP2)*AJ*((CV(IP2)*CV(JP2)+SV(IP2)*SV(JP2))*
01220      1  T(IP7)*T(JP7)-TP(IP7)*TP(JP7)/BK/BK/RHOC1(JR2))*
01230      2  TCT(JR7)*G2(J4)
01240       Z2=Z2+DH(IP2)*AJ*TZ(IP7)*TP(JP7)*TCT(JR7)/RHOC1(JR2)*
01250      1  G2(J4)
01260       Z3=Z3+DH(IP2)*AJ*((CV(IP2)*CPC(JP2)+SV(IP2)*SPC(JP2))*
01270      1  T(IP7)*TCR(JR7)-TP(IP7)*(TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)
01280      2  /BK/BK)*TZ(JP7)*G2(J4)
01290       Z4=Z4+DH(IP2)*AJ*TZ(IP7)*TZ(JP7)*
01300      1  (TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)*G2(J4)
01332 71    CONTINUE
01342 70    CONTINUE
01350       Z(L1)=U*C1*Z1
01360       Z(L2)=-M*C2*Z2
01370       Z(L3)=U*C3*Z3
01380       Z(L4)=-M*C4*Z4
01392 31    CONTINUE
01402 30    CONTINUE
01404       IF(NE.EQ.0) GO TO 900
01410C BOT-EDGE LOOP.
01420       DO 130 JC=1,NC
01430       DO 130 JP=1,NM
01450       L3=NM2*LC2+((JC-1)*NM+(JP-1))*NM2
01480       L4=L3+NM
01490       DO 131 IP=1,NM
01520       L3=L3+1
01530       L4=L4+1
01560       Z3=0.0
01570       Z4=0.0
01572       Z3E=0.0
01574       Z4E=0.0
01580       JP2=2*(JP-1)
01590       JP7=4*(JP-1)
01600       DO 170 JJP=1,4
```

```
C1610       JP2=JP2+1
C1620       JP7=JP7+1
C1630       JR2=2*LP
C1640       JR7=2*(JC-1)
C1642       JZ2=2*(JC-1)
C1644       JZ7=2*(JC-1)
C1650       DO 170 JJRZ=1,2
C1660       JR2=JR2+1
C1670       JR7=JR7+1
C1672       JZ2=JZ2+1
C1674       JZ7=JZ7+1
C1680       IP2=2*(IP-1)
C1690       IP7=4*(IP-1)
C1700       DO 171 IIP=1,4
C1710       IP2=IP2+1
C1720       IP7=IP7+1
C1730       AJ=AC(JP2)*ABS(RHOC(JR2+1)**2-RHOC(JR2)**2)
C1740       RJ=RHOC1(JR2)*RC1(JP2)
C1750       J4=((JC-1)*NP1*NPR1+(JP2-1)*NPR1+(JR2-1))*NP1+IP2
C1760C BOT-EDGE(CAP) IMPEDANCE ELEMENTS COMPUTED.
C1770C Z3 = ZTR , Z4 = ZZR.
C1830       Z3=Z3+DH(IP2)*AJ*((CV(IP2)*CPC(JP2)+SV(IP2)*SPC(JP2))*
C1840      1 T(IP7)*TCE(JR7)-TP(IP7)*(TPCE(JR7)/RC1(JP2)+TCE(JR7)/RJ)
C1850      2 /BK/BK)*TZ(JP7)*G2(J4)
C1860       Z4=Z4+DH(IP2)*AJ*TZ(IP7)*TZ(JP7)*
C1870      1 (TPCE(JR7)/RC1(JP2)+TCE(JR7)/RJ)*G2(J4)
C1940       J4=((JC-1)*NP1*2+(JP2-1)*2+(JJRZ-1))*NP1+IP2
C1981       AJ=DH(JP2)*ABS(ZE(JC)-ZC(JC))/2.
C1982C BOT-EDGE(BOT) IMPEDANCE ELEMENTS COMPUTED.
C1983C Z(L3) = ZTZ , Z(L4) = ZZZ.
C1984       Z3E=Z3E+DH(IP2)*AJ*TP(IP7)*TZ(JP7)*TPBE(JZ7)*G2E(J4)
C1985       Z4E=Z4E+DH(IP2)*AJ*TZ(IP7)*TZ(JP7)*(TBE(JZ7)*
C1986      1 (G2E(J4)-(-1)**M*G1E(J4))+U*M*PI/BK/BKL*TPBE(JZ7)*G2E(J4))
C1987 171   CONTINUE
C1988 170   CONTINUE
C1991       Z(L3)=U*C3*Z3-U*C3E*Z3E
C1992       Z(L4)=-M*C4*Z4+U*C4E*Z4E
C1993 131   CONTINUE
C1994 130   CONTINUE
C1995C **** END CALCULATION OF ZSC MATRIX ****
C1996 900   CONTINUE
C1998C PRINT OUT IMPEDANCE MATRIX Z(M) BY SUBMATRIX BY COLUMNS,
C2000       LI(1)=NM
C2001       LI(2)=NP
C2002       LJ(1)=LC
C2003       LJ(2)=LC
C2004       LJ(3)=LE
C2005       CALL ZLIST(2,3,LI,LJ,NAMI,NAMJ,Z)
C2012C WRITE ZSC TO DISK.
C2014       WRITE(7) M
C2016       WRITE(7) (Z(I),I=1,LSC)
C2020 998   CONTINUE
```

```
C2030       STOP
C2040       END
C2050C ------------------------------------------------------------
C2055       SUBROUTINE BOTIN
C2060C
C2065C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
C2070C
C2075       COMMON /BOT1/ NMODE,NPT,NBAND
C2080       COMMON /BOT2/ NP,BL,YR(83),YB(83),YB1(82),XB1(82)
C2085       COMMON /BOT3/ DH(82),SV(82),CV(82)
C2090       COMMON /BOT5/ T(160),TP(160),TZ(160)
C2095       COMMON /BOT6/ IEDGE,IUNIF
C2100       READ(5,49) NMODE,NPT,NBAND
C2105       READ(5,49) NP
C2110  49   FORMAT(3I3)
C2115       WRITE(6,48) NMODE,NPT,NBAND,NP
C2120  48   FORMAT(32H     NMODE    NPT    NBAND      NP,/,4I8)
C2125       READ(5,53)(YB(I),I=1,NP)
C2130       READ(5,53)(XB(I),I=1,NP)
C2135    53 FORMAT(10F8.4)
C2140       WRITE(6,55)
C2145  55   FORMAT(/,3H YB)
C2150       WRITE(6,46)(YB(I),I=1,NP)
C2155    46 FORMAT(1X,10F8.4)
C2160       WRITE(6,56)
C2165  56   FORMAT(/,3H XB)
C2170       WRITE(6,46)(XB(I),I=1,NP)
C2175C PLOT THE BODY COORDINATES.
C2180       CALL PLOTB(XB,YB,NP,41)
C2185       READ(5,53) BL
C2190       WRITE(6,47) BL
C2195  47   FORMAT(//,21H HALF-LENGTH OF BOT =,F12.4)
C2200C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
C2205       IF(MOD(NP,2).NE.1) GO TO 980
C2210C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
C2215       IEDGE=1
C2220       IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
C2225       IEDGE=0
C2230       YB(NP+1)=YB(2)
C2235       XB(NP+1)=XB(2)
C2240       YB(NP+2)=YB(3)
C2245       XB(NP+2)=XB(3)
C2250       NP=NP+2
C2255       WRITE(6,66) NP
C2260  66   FORMAT(/,39H BOT GENERATING CURVE IS CLOSED.  NP = ,I3)
C2265  10   CONTINUE
C2270C COMPUTATION OF BODY SEGMENT PARAMETERS.
C2275       DO 57 I=2,NP
C2280       I2=I-1
C2285       RR1=YB(I)-YB(I2)
C2290       RR2=XB(I)-XB(I2)
C2295       DH(I2)=SQRT(RR1*RR1+RR2*RR2)
```

```
C2300        XB1(I2)=.5*(XB(I)+XB(I2))
C2305        YB1(I2)=.5*(YB(I)+YB(I2))
C2310        SV(I2)=RR1/DH(I2)
C2315        CV(I2)=RR2/DH(I2)
C2320  57    CONTINUE
C2325C CHECK IF BOT SEGMENTATION IS UNIFORM.
C2330        IUNIF=0
C2335        NP1=NP-1
C2340        DO 60 I=2,NP1
C2345        RR1=DH(I)/DH(1)
C2350        IF(RR1.LT.0.99 .OR. RR1.GT.1.01) GO TO 20
C2355  60    CONTINUE
C2360        IUNIF=1
C2365        WRITE(6,67)
C2370  67    FORMAT(//,' BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
C2375  20    CONTINUE
C2380C COMPUTATION OF TRIANGLE FUNCTIONS T.
C2385        NM=(NP-3)/2
C2390        DO 74 J=1,NM
C2395        J2=2*(J-1)+1
C2400        J3=J2+1
C2405        J4=J3+1
C2410        J5=J4+1
C2415        J6=4*(J-1)+1
C2420        J7=J6+1
C2425        J8=J7+1
C2430        J9=J8+1
C2435        DEL1=DH(J2)+DH(J3)
C2440        DEL2=DH(J4)+DH(J5)
C2445        TP(J6)=1./DEL1
C2450        TP(J7)=1./DEL1
C2455        TP(J8)=-1./DEL2
C2460        TP(J9)=-1./DEL2
C2465        T(J6)=DH(J2)/2./DEL1
C2470        T(J7)=(DH(J2)+DH(J3)/2.)/DEL1
C2475        T(J8)=(DH(J4)/2.+DH(J5))/DEL2
C2480        T(J9)=DH(J5)/2./DEL2
C2485    74  CONTINUE
C2490        NM4=NM*4
C2495        DO 75 J=1,NM4
C2500  75    TZ(J)=T(J)
C2505        IF(IEDGE.EQ.0) GO TO 76
C2510        TZ(1)=2.-T(1)
C2515        TZ(2)=2.-T(2)
C2520        TZ(NM4-1)=2.-T(NM4-1)
C2525        TZ(NM4)=2.-T(NM4)
C2530  76    CONTINUE
C2535        RETURN
C2540  980   WRITE(6,981)
C2545  981   FORMAT(//,' **** ERROR IN BOT INPUT')
C2550        STOP
C2555        END
```

```
C3090C  ************************************************************
C3100         SUBRCUTINE PLOTR(X,Y,N,NR)
C3110C  ************************************************************************
C3120C
C3130C  WRITTEN 2/14/74   BY J. M. PUTNAM        DEPT 220      X23877
C3140C
C3150C  THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C3160C     N IS THE NUMBER OF POINTS TO BE PLOTTED.
C3170C     NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C3180C     NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C3190C        NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C3200C
C3210         REAL X(1),Y(1),HEAD(10)
C3220         INTEGER  LINE(101),BLANK,STAR,PLUS
C3230         DATA BLANK,STAR,PLUS /1H ,1H*,1H+/
C3240         NC=51
C3250         N10=(NC-1)/10
C3260         WRITE(6,500)
C3270  500    FORMAT(//,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
C3280         WRITE(6,504)
C3290         XMIN=X(1)
C3300         XMAX=X(1)
C3310         YMIN=Y(1)
C3320         YMAX=Y(1)
C3330         DO 6 I=1,N
C3340         IF(X(I).LT.XMIN) XMIN=X(I)
C3350         IF(X(I).GT.XMAX) XMAX=X(I)
C3360         IF(Y(I).LT.YMIN) YMIN=Y(I)
C3370         IF(Y(I).GT.YMAX) YMAX=Y(I)
C3380  6      CONTINUE
C3390         DEL=XMAX-XMIN
C3400         IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C3410         XMAX=XMIN+DEL
C3420         YMAX=YMIN+DEL
C3430         DO 5 I =1,N10
C3440         Z=I
C3450      5  HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
C3460         DY=(YMAX-YMIN)/(NR-1)
C3470         IEDGE=1
C3480         IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
C3490         Z=YMAX+DY
C3500         YL=Z-DY/2.
C3510         DO 7 J=1,NR
C3520         DO 8 K=1,NC
C3530  8      LINE(K)=BLANK
C3540         Z=Z-DY
C3550         YU=YL
C3560         YL=Z-DY/2.
C3570         DO 9 I=1,N
C3580         IF(Y(I).GE.YU) GO TO 9
C3590         IF(Y(I).LT.YL) GO TO 9
C3600         K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
```

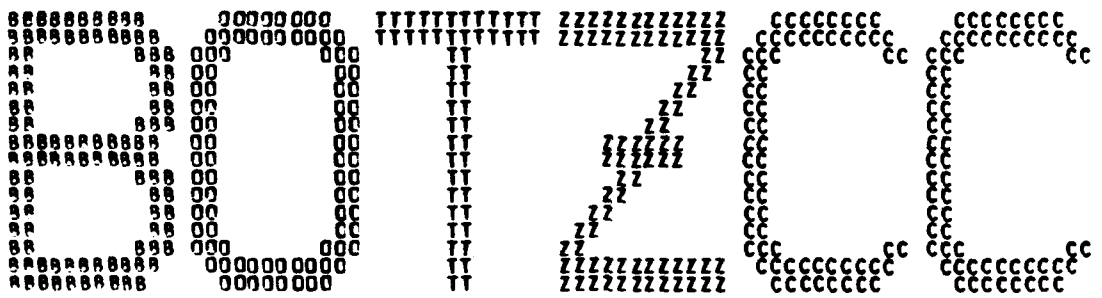
```
C3610       IF(K.GT.NC) K=NC
C3620       LINE(K)=STAR
C3630       IF(MOD(I,2).EQ.1) LINE(K)=PLUS
C3640       IF(IEDGE.EQ.0) GO TO 9
C3650       IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
03660  9    CONTINUE
C3670       WRITE(6,508) Z,(LINE(K),K=1,NC)
C3680  7    CONTINUE
C3690       WRITE(6,504)
C3700       WRITE( 6,306?)
C3710       WRITE( 6,507) XMIN,(HEAD(I),I=1,N10)
C3720       RETURN
C3730  504 FORMAT ( 1X, 14(1H-), 1H., 10(5H----.), 1H- )
03740  507 FORMAT(10X,11(F10.4))
03750  508 FORMAT (1X, F12.4,1X, 1HI, 51A1, 1HI )
03760 3002 FORMAT(4X,7HYH / XH,4X,1HI,5(9X,1HI))
C3770       END
03780C -------------------------------------------------
03785       SUBROUTINE CAPIN
C3790C
C3795C READ CAP INPUTS AND COMPUTE CAP ARRAYS.
C3800C
C3805       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C3810       COMMON /CAP1/ NC,XC,YC,ZC(2)
C3815       COMMON /CAP2/ NPR,RHOC(21),PHOC1(20),DRHOC(20)
03820       COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
03825       COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
C3830       COMMON /EDG1/ NE,ZE(2),ZBE(10)
C3835       COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
C3840       READ(5,1) NC,NPR,NE
C3845  1    FORMAT(3I3)
C3850       IF(NE.NE.0) NE=NC
C3855       WRITE(6,3) NC,NPR,NE
C3860  3    FORMAT(16H1   NC   NPR   NE,//,3I5,///)
C3865       IF(NC.EQ.0) RETURN
C3870       READ(5,2) XC,YC
C3875  2    FORMAT(10F8.4)
C3880       READ(5,2) (ZC(I),I=1,NC)
C3885       READ(5,2) (RHOC(I),I=1,NPR)
C3890       IF(NE.NE.0) READ(5,2) (ZE(I),I=1,NE)
C3895       WRITE(6,4)
C3900  4    FORMAT(37H  CAP       XC       YC       ZC       ZE,/)
C3905       DO 10C I=1,NC
C3910       IF(NE.EQ.0) WRITE(6,5) I,XC,YC,ZC(I)
C3915       IF(NE.NE.0) WRITE(6,5) I,XC,YC,ZC(I),ZE(I)
03920 10C   CONTINUE
C3925  5    FORMAT(I4,4X,4F8.4)
C3930       WRITE(6,6)
C3935  6    FORMAT(//,5H RHOC)
C3940       WRITE(6,7) (RHOC(I),I=1,NPR)
C3945  7    FORMAT(1X,10F8.4)
C3950       IF(MOD(NPR,2).NE.1) GO TO 990
```

```
C3955        DO 120 I=2,NPR
C3960  120   IF(RHOC(I).LE.RHOC(I-1)) GO TO 980
C3965C COMPUTATION OF CAP SECTOR PARAMETERS.
C3970        DO 47 I=1,NP
C3975        RC(I)=SQRT((YB(I)-YC)**2+(XB(I)-XC)**2)
C3980  47    CONTINUE
C3985        DO 57 I=2,NP
C3990        I2=I-1
C3995        RR1=YB1(I2)-YC
C4000        RR2=XB1(I2)-XC
C4005        RC1(I2)=SQRT(RR1*RR1+RR2*RR2)
C4010        AC(I2)=ABS((XB(I)-XC)*(YB(I)+YC)+(XB(I2)-XB(I))*
C4015       1  (YB(I2)+YB(I))+(XC-XB(I2))*(YC+YB(I2)))/2.
C4020        SPC(I2)=RR1/RC1(I2)
C4025        CPC(I2)=RR2/RC1(I2)
C4030  57    CONTINUE
C4035        DO 67 I=2,NPR
C4040        I2=I-1
C4045        RHOC1(I2)=(RHOC(I)+RHOC(I2))/2.
C4050        DRHOC(I2)=RHOC(I)-RHOC(I2)
C4055  67    CONTINUE
C4060C COMPUTATION OF CAP TRIANGLE FUNCTIONS.
C4065        LC=(NPR-3)/2
C4070        DO 74 J=1,LC
C4075        J2=2*(J-1)+1
C4080        J3=J2+1
C4085        J4=J3+1
C4090        J5=J4+1
C4095        J6=4*(J-1)+1
C4100        J7=J6+1
C4105        J8=J7+1
C4110        J9=J8+1
C4115        DEL1=DRHOC(J2)+DRHOC(J3)
C4120        DEL2=DRHOC(J4)+DRHOC(J5)
C4125        TPCR(J6)=1./DEL1
C4130        TPCR(J7)=1./DEL1
C4135        TPCR(J8)=-1./DEL2
C4140        TPCR(J9)=-1./DEL2
C4145        TCR(J6)=DRHOC(J2)/2./DEL1
C4150        TCR(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
C4155        TCR(J8)=(DRHOC(J4)/2.+DRHOC(J5))/DEL2
C4160        TCR(J9)=DRHOC(J5)/2./DEL2
C4165    74  CONTINUE
C4170        LC4=LC*4
C4175        DO 75 I=1,LC4
C4180        TCI(I)=TCR(I)
C4185        TPCT(I)=TPCR(I)
C4190  75    CONTINUE
C4195        TCT(LC4-1)=2.-TCT(LC4-1)
C4200        TCI(LC4)=2.-TCI(LC4)
C4205        TPCT(LC4-1)=-TPCT(LC4-1)
C4210        TPCT(LC4)=-TPCT(LC4)
```

```
C4215       IF(RHOC(1).EQ.0.0) GO TO 76
C4220       TCT(1)=2.-TCT(1)
C4225       TCT(2)=2.-TCT(2)
C4230       TPCT(1)=-TPCT(1)
C4235       TPCT(2)=-TPCT(2)
C4240  76   CONTINUE
C4245       IF(NE.EQ.0) RETURN
C4250C COMPUTATION OF EDGE HALF TRIANGLE FUNCTIONS.
C4255       DO 8C IC=1,NC
C4260       J2=NPR-2
C4265       J3=J2+1
C4270       J6=2*(IC-1)+1
C4275       J7=J6+1
C4280       DEL1=DRHOC(J2)+DRHOC(J3)
C4285       TPCE(J6)=1./DEL1
C4290       TPCE(J7)=1./DEL1
C4295       TCE(J6)=DRHOC(J2)/2./DEL1
C4300       TCE(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
C4305       DEL2=ZE(IC)-ZC(IC)
C4310       ZBE(J6)=ZC(IC)+0.25*DEL2
C4315       ZBE(J7)=ZC(IC)+0.75*DEL2
C4320       IF(DEL2.LT.0.0) GO TO 78
C4325C EDGE IS AT Z=-L.
C4330       TPBE(J6)=-1./DEL2
C4335       TPBE(J7)=-1./DEL2
C4340       TBE(J6)=0.75
C4345       TBE(J7)=0.25
C4350       GO TO 80
C4355  78   CONTINUE
C4360C EDGE IS AT Z=+L.
C4365       TPBE(J6)=1./DEL2
C4370       TPBE(J7)=1./DEL2
C4375       TBE(J6)=-0.75
C4380       TBE(J7)=-0.25
C4385  8C   CONTINUE
C4390       RETURN
C4395  980  WRITE(6,981)
C4400  981  FORMAT(//,' **** ERROR IN CAP INPUT')
C4405       STOP
C4410       END
C4910C ********************************************************************
C4920       SUBROUTINE CSIMP(F,A,B,DEL,IMAX,SI1,S,N,IER)
C4930C ********************************************************************
C4940C COMPLEX SIMPSON INTEGRATION ROUTINE.
C4950       COMPLEX F,SI1,S,SUMK
C4960       SI1=(1.0,.0)
C4970       S=(.0,.0)
C4980       N=0
C4990       BA=B-A
C5000       IF (BA)20,19,20
C5010   19  IER=1
C5020       RETURN
```

```
C5030      20 IF(DEL)22,22,23
C5040      22 IER=2
C5050         RETURN
C5060      23 IF(IMAX-1)24,24,25
C5070      24 IER=3
C5080         RETURN
C5090      25 X=BA/2.+A
C5100         NHALF=1
C5110         SUMK=F(X)*BA*2./3.
C5120         S=SUMK+(F(A)+F(B))*BA/6.
C5130         DO 28 I=2,IMAX
C5140         SI1=S
C5150         S=(S-SUMK/2.)/2.
C5160         NHALF=NHALF*2
C5170         ANHLF=NHALF
C5180         FRSTX=A+(BA/ANHLF)/2.
C5190         SUMK=F(FRSTX)
C5200         XK=FRSTX
C5210         KLAST=NHALF-1
C5220         FINC=BA/ANHLF
C5230         DO 26 K=1,KLAST
C5240         XK=XK+FINC
C5250      26 SUMK=SUMK+F(XK)
C5260         SUMK=SUMK*2.*BA/(3.*ANHLF)
C5270         S=S+SUMK
C5280         IF(CABS(S).EQ.0.0) GO TO 29
C5290         IF((CABS(S-SI1)/CABS(S))-DEL) 29,28,28
C5300      28 CONTINUE
C5310         IER=4
C5320         WRITE(6,1) SI1,S
C5330  1      FORMAT(//,29H INTEGRATION DID NOT CONVERGE,//
C5340        1 50H THE PREVIOUS AND FINAL VALUES OF THE INTEGRAL ARE,
C5350        2 13H RESPECTIVELY,4E12.3)
C5360         GO TO 30
C5370      29 IER=6
C5380      30 M=2*NHALF
C5390         RETURN
C5400         END
C5410C ********************************************************
C5412         COMPLEX FUNCTION FUNC1(E)
C5414C ********************************************************
C5416         COMPLEX U
C5418         COMMON /WAVE/ BK
C5420         COMMON /INT/ M,RHO2,ZP
C5422         DATA PI,L /3.14159265,(0.,1.)/
C5424         R=SQRT(RHO2+(E-ZP)**2)
C5426         FUNC1=CEXP(-U*BK*R)/R
C5428         RETURN
C5430         END
C5432C ********************************************************
C5434         COMPLEX FUNCTION FUNC2(E)
C5436C ********************************************************
```

```
      COMPLEX U
      COMMON /WAVE/ BK
      COMMON /BOT2/ ZP,PL
      COMMON /INT/ M,RHO2,ZP
      DATA PI,U /3.14159265,(0.,1.)/
      R=SQRT(RHO2+(E-ZP)**2)
      FUNC2=CEXP(-U*(M*PI*E/BL+BK*R))/R
      RETURN
      END
C-----------------------------------------------------
      SUBROUTINE ZLIST(NI,NJ,LI,LJ,NAMI,NAMJ,Z)
C WRITE Z TO THE LINE-PRINTER BY SUBMATRIX BY COLUMNS.
      COMPLEX Z(1)
      DIMENSION LI(1),LJ(1),NAMI(1),NAMJ(1)
      NCOL=0
      DO 100 I=1,NI
100   NCOL=NCOL+LI(I)
      NR=1
      DO 300 J=1,NJ
      IF(LJ(J).EQ.0) GO TO 300
      NC=1
      DO 200 I=1,NI
      IF(LI(I).EQ.0) GO TO 200
      WRITE(6,1) NAMI(I),NAMJ(J)
1     FORMAT(/,2H Z,1X,A4,3H - ,A4)
      KI=LI(I)
      KJ=LJ(J)
      K1=(NR-1)*NCOL+NC
      DO 150 KK=1,KJ
      K2=K1+KI-1
      WRITE(6,2) (Z(K),K=K1,K2)
2     FORMAT(1X,10G11.4)
150   K1=K1+NCOL
      NC=NC+LI(I)
200   CONTINUE
      NR=NR+LJ(J)
300   CONTINUE
      RETURN
      END
/ECR
```



```
00100BOTZ,P30,T1CO,CM144000,STCX3.
00110ACCOUNT,MO326S,BOTZA.
00120BANNERS(OUTPUT) *J, *OUTNAM*DEPT 220*BLD 110-4**
00130DEFINE(OUTFIL=ZCCX)
00138FTN(R=0,OPT=2)
00140LDSET(PRESET=INDEF)
00150LGO.
00210EXIT.
00230/EOR
00240       PROGRAM BOTZCC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
00250      1 OUTFIL,TAPE7=OUTFIL)
00252C
00254C BOTZCC GENERATES THE CAP Z MATRIX.
00256C
00258C UNIT 5 IS THE CARD READER.
00260C UNIT 6 IS THE LINE PRINTER.
00262C UNIT 7 IS A DISK FILE FOR OUTPUT OF THE Z MATRIX.
00264C
00280       COMPLEX U,A3,Z(9216),GS(1640)
00281       DIMENSION TWGHT(360),RWGHT(360),EWGHT(40)
00282       COMPLEX Z1,Z2,Z3,Z4
00283       COMPLEX Z3E,Z4E
00284       COMMON /WAVE/ BK
00285       COMMON /BOT1/ NMODE,NPT,NBAND
00287       COMMON /BOT2/ NP,RL,YR(83),XB(83),YB1(82),XB1(82)
00288       COMMON /BOT3/ DH(82),SV(82),CV(82)
00289       COMMON /BOT5/ T(160),TP(160),TZ(160)
00290       COMMON /CAP1/ NC,XC,YC,ZC(2)
00292       COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
00294       COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
00296       COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
00298       COMMON /EDG1/ NE,ZE(2),ZRE(10)
00300       COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
00320       COMMON /INT/ CONST1,CONST2
00325       DIMENSION LI(3),LJ(3),NAMI(3),NAMJ(3)
00330       EXTERNAL FUNC
00340       DATA NAMI,NAMJ /4HC(T),4HC(R),4HEDGE,4HC(T),4HC(R),4HEDGE/
00430       U=(0.,1.)
00432       PI=3.14159265
00434       ETA=376.707
00435       READ(5,1) BK
00436  1    FORMAT(E15.7)
00437       WRITE(6,2) BK
00438  2    FORMAT(9H1       BK=,/,E15.7)
00440C
00450       CALL BOTIN
00460C
00470       CALL CAPIN
00480C
00485       WRITE(7) NC,NPR,NE
00490       NR=(NP-3)/2
00500       LR=(NPR-3)/2
```

```
00510       LC=NC*NM*LR
00520       LC2=LC*2
00525       LE=NE*NM
00530       LCC=(LC2+LE)**2
00540       NP1=NP-1
00550       NPR1=NPR-1
00552C CALCULATE IMPEDANCE LOADING WEIGHTS ON CAP.
00554       J=0
00556       DO 20 JP=1,NM
00558       J2=2*(JP-1)+1
00560       J3=J2+1
00562       J4=J3+1
00564       J5=J4+1
00566       J6=4*(JP-1)+1
00568       J7=J6+1
00570       J8=J7+1
00572       J9=J8+1
00574       DO 10 JR=1,LR
00576       J=J+1
00578       J1=2*(JR-1)+1
00580       TWGHT(J)=(RHOC(J1+4)**2-RHOC(J1)**2)*
00582      1 (AC(J2)*T(J6)+AC(J3)*T(J7)+AC(J4)*T(J8)+AC(J5)*T(J9))
00584       RWGHT(J)=(RHOC(J1+4)**2-RHOC(J1)**2)*(AC(J2)*TZ(J6)+
00586      1 AC(J3)*TZ(J7)+AC(J4)*TZ(J8)+AC(J5)*TZ(J9))
00588 10    CONTINUE
00590       IF(NE.EQ.0) GO TO 20
00592       EWGHT(JP)=(RHOC(J1+4)**2-RHOC(J1+2)**2)*(AC(J2)*TZ(J6)+
00594      1 AC(J3)*TZ(J7)+AC(J4)*TZ(J8)+AC(J5)*TZ(J9))+
00596      2 ABS(ZE(1)-ZC(1))*(DH(J2)*TZ(J6)+DH(J3)*TZ(J7)+
00598      3 DH(J4)*TZ(J8)+DH(J5)*TZ(J9))
00600 20    CONTINUE
00602       I1=NM*LR
00604       WRITE(7) (TWGHT(I),I=1,I1)
00606       WRITE(7) (RWGHT(I),I=1,I1)
00608       IF(NE.NE.0) WRITE(7) (EWGHT(I),I=1,NM)
00610C COMPUTATION OF GREEN'S FUNCTION KERNEL SELF TERM.
00740       I1=0
00742       NPTI=0
00744       NINTI=0
00746       DO 17 JP=1,NP1
00748       DO 17 JR=1,NPR1
00750       I1=I1+1
00770C SELF TERM
00790       R1=RC1(JP)*RHOC(JR)
00800       R2=RC1(JP)*RHOC(JR+1)
00810       C2TH1=-(DH(JP)**2-RC(JP+1)**2-RC(JP)**2)/2./
00820      1 RC(JP+1)/RC(JP)
00830       TH1=ACOS(C2TH1)/2.
00840       CONSI1=(R1+R2)**2/4.
00850       CONST2=-(R1+R2)*COS(TH1)
00860       CALL CSIMP(FUNC,R1,R2,0.05,10,A3,GS(I1),K,IER)
00870       NPTI=NPTI+K+1
```

```
C0880       NINT1=NINT1+1
C0890       GS(I1)=GS(I1)*(R1+R2)*(R2-R1)*TH1*TH1/2./PI
C0900 17    CONTINUE
C0910       AVG1=NPT1/FLOAT(NINT1)
C0920       WRITE(6,56) NINT1,AVG1
C0930 56    FORMAT(/,' NUMBER OF INTEGRATIONS AND AVERAGE NUMBER OF',
C0940      1 ' POINTS PER INTEGRATION FOR SELF TERM',/,(I8,G10.3))
C0950C *** END CALCULATION OF GREEN'S KERNEL ***
C0951C CAP-CAP LOOP.
C0952       DO 30 JC=1,NC
C0954       DO 30 JP=1,NM
C0956       DO 30 JR=1,LR
C0958       L1=((JC-1)*NM*LR+(JP-1)*LR+(JR-1))*(LC2+LE)
C0960       L2=L1+LC
C0962       L3=L1+LC*(LC2+LE)
C0964       L4=L3+LC
C0966       DO 31 IC=1,NC
C0968       DO 31 IP=1,NM
C0970       DO 31 IR=1,LR
C0972       L1=L1+1
C0974       L2=L2+1
C0976       L3=L3+1
C0978       L4=L4+1
C0980       Z1=0.0
C0982       Z2=0.0
C0984       Z3=0.0
C0986       Z4=0.0
C0990       JP2=2*(JP-1)
C0992       JP7=4*(JP-1)
C0994       DO 70 JJP=1,4
C0996       JP2=JP2+1
C0998       JP7=JP7+1
C1000       JR2=2*(JR-1)
C1002       JR7=4*(JR-1)
C1004       DO 70 JJR=1,4
C1006       JR2=JR2+1
C1008       JR7=JR7+1
C1010       IP2=2*(IP-1)
C1012       IP7=4*(IP-1)
C1014       DO 71 IIP=1,4
C1016       IP2=IP2+1
C1018       IP7=IP7+1
C1020       IR2=2*(IR-1)
C1022       IR7=4*(IR-1)
C1024       DO 71 IIR=1,4
C1026       IR2=IR2+1
C1028       IR7=IR7+1
C1030       IF(IC.NE.JC) GO TO 60
C1031       IF(IR2.NE.JR2) GO TO 60
C1032       IF(YB1(IP2).NE.Y91(JP2)) GO TO 60
C1033       IF(XB1(IP2).NE.XB1(JP2)) GO TO 60
C1034C SELF TERM.
```

```
C1035       A3=GS(IR2+(IP2-1)*NPR1)
C1036       GO TO 65
C1037C NON-SELF TERM.
C1038 60    CONTINUE
C1042       XX=RHOC1(JR2)*RC1(JP2)*CPC(JP2)-RHOC1(IR2)*RC1(IP2)*CPC(IP2)
C1052       YY=RHOC1(JR2)*RC1(JP2)*SPC(JP2)-RHOC1(IR2)*RC1(IP2)*SPC(I*2)
C1062       ZZ=ZC(JC)-ZC(IC)
C1072       RHO2=XX*XX+YY*YY+ZZ*ZZ
C1102       AJ=AC(JP2)*ABS(RHOC(JR2+1)**2-RHOC(JR2)**2)
C1112       AI=AC(IP2)*ABS(RHOC(IR2+1)**2-RHOC(IR2)**2)
C1122       RR=SQRT(RHO2)
C1132       A3=AI*AJ*CEXP(-U*BK*RR)/4./PI/RR
C1182 65    CONTINUE
C1184       RI=RHOC1(IR2)*RC1(IP2)
C1186       RJ=RHOC1(JR2)*RC1(JP2)
C1192C CAP-CAP IMPEDANCE ELEMENTS COMPUTED.
C1202C Z(L1) = ZTT , Z(L2) = ZRT , Z(L3) = ZTR , Z(L4) = ZRR.
C1212       Z1=Z1+TCT(IR7)*TCT(JR7)*((CV(IP2)*CV(JP2)+
C1222      1 SV(IP2)*SV(JP2))*T(IP7)*T(JP7)
C1232      2 -TP(IP7)*TP(JP7)/BK/BK/RHOC1(IR2)/RHOC1(JR2))*A3
C1242       Z2=Z2+TZ(IP7)*TCT(JR7)*((CPC(IP2)*CV(JP2)+
C1252      1 SPC(IP2)*SV(JP2))*TCR(IR7)*T(JP7)
C1262      2 -(TPCR(IR7)/RC1(IP2)+TCR(IR7)/RI)*
C1264      3 TP(JP7)/RHOC1(JR2)/BK/BK)*A3
C1272       Z3=Z3+TCT(IR7)*TZ(JP7)*((CV(IP2)*CPC(JP2)+
C1282      1 SV(IP2)*SPC(JP2))*TZ(IP7)*TCR(JR7)
C1292      2 -TP(IP7)/RHOC1(IR2)*
C1294      3 (TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)/BK/BK)*A3
C1302       Z4=Z4+TZ(IP7)*TZ(JP7)*((CPC(IP2)*CPC(JP2)+
C1312      1 SPC(IP2)*SPC(JP2))*TCR(IR7)*TCR(JR7)
C1322      2 -(TPCR(IR7)/RC1(IP2)+TCR(IR7)/RI)*
C1324      3 (TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)/BK/BK)*A3
C1332 71    CONTINUE
C1342 70    CONTINUE
C1352       Z(L1)=U*BK*ETA*Z1
C1362       Z(L2)=U*BK*ETA*Z2
C1372       Z(L3)=U*BK*ETA*Z3
C1382       Z(L4)=U*BK*ETA*Z4
C1392 31    CONTINUE
C1402 30    CONTINUE
C1413       IF(NE.EQ.0) GO TO 900
C1404C CAP-EDGE LOOP.
C1412       DO 130 JC=1,NC
C1422       DO 130 JP=1,NM
C1462       L3=(LC2+LE)*LC2+((JC-1)*NM+(JP-1))*(LC2+LE)
C1472       L4=L3+LC
C1482       DO 131 IC=1,NC
C1492       DO 131 IP=1,NM
C1502       DO 131 IR=1,LP
C1532       L3=L3+1
C1542       L4=L4+1
C1572       Z3=0.0
```

```
01582         Z4=0.0
01584         Z3E=C.0
01586         Z4E=C.0
01592         JP2=2*(JP-1)
01602         JP7=4*(JP-1)
01612         DO 170 JJP=1,4
01622         JP2=JP2+1
01632         JP7=JP7+1
01642         JR2=2*LR
01652         JR7=2*(JC-1)
01654         JZ2=2*(JC-1)
01656         JZ7=2*(JC-1)
01662         DO 170 JJRZ=1,2
01672         JR2=JR2+1
01682         JR7=JR7+1
01684         JZ2=JZ2+1
01686         JZ7=JZ7+1
01692         IP2=2*(IP-1)
01702         IP7=4*(IP-1)
01712         DO 171 IIP=1,4
01722         IP2=IP2+1
01732         IP7=IP7+1
01742         IR2=2*(IR-1)
01752         IR7=4*(IR-1)
01762         DO 171 IIR=1,4
01772         IR2=IR2+1
01782         IR7=IR7+1
01784         AIC=AC(IP2)*ABS(RHOC(IR2+1)**2-RHOC(IR2)**2)
01786         AJC=AC(JP2)*ABS(RHOC(JR2+1)**2-RHOC(JR2)**2)
01788         AJB=CH(JP2)*ABS(ZE(JC)-ZC(JC1)/2.
01792         IF(IC.NE.JC) GO TO 160
01802         IF(IR2.NE.JR2) GO TO 160
01812         IF(YB1(IP2).NE.YB1(JP2)) GO TO 160
01822         IF(XB1(IP2).NE.XB1(JP2)) GO TO 160
01832C SELF TERM.
01842         A3=GS(IR2+(IP2-1)*NPR1)
01852         GO TO 165
01862C NON-SELF TERM.
01872 160     CONTINUE
01882         XX=RHOC1(JR2)*RC1(JP2)*CPC(JP2)-RHOC1(IR2)*RC1(IP2)*CPC(IP2)
01892         YY=RHOC1(JR2)*RC1(JP2)*SPC(JP2)-RHOC1(IR2)*RC1(IP2)*SPC(IP2)
01902         ZZ=ZC(JC)-ZC(IC)
01912         RHO2=XX*XX+YY*YY+ZZ*ZZ
01942         RR=SQRT(RHO2)
01952         A3=AIC*AJC*CEXP(-U*BK*RR)/4./PI/RR
01962 165     CONTINUE
01972         RI=RHOC1(IR2)*RC1(IP2)
01982         RJ=RHOC1(JR2)*RC1(JP2)
01992C CAP-EDGE(CAP) IMPEDANCE ELEMENTS COMPUTED.
02002C Z(L3) = ZTR , Z(L4) = ZRR.
02082         Z3=Z3+TC(IP7)*TZ(JP7)*((CV(IP2)*CPC(JP2)+
02092        1 SV(IP2)*SPC(JP2))*T(IP7)*TCE(JR7)
```

```
C2102       2 -TP(IP7)/RHOC1(IR2)*
C2112       3 (TPCE(JR7)/RC1(JP2)+TCE(JR7)/RJ)/BK/BK)*A3
C2122         Z4=Z4+TZ(IP7)*TZ(JP7)*((CPC(IP2)*CPC(JP2)+
C2132       1 SPC(IP2)*SPC(JP2))*TCR(IR7)*TCE(JR7)
C2142       2 -(TPCR(IR7)/RC1(IP2)+TCR(IR7)/RI)*
C2152       3 (TPCE(JR7)/RC1(JP2)+TCE(JR7)/RJ)/BK/BK)*A3
C2154         XX=XB1(JP2)-(XC+RHOC1(IR2)*RC1(IP2)*CPC(IP2))
C2156         YY=YB1(JP2)-(YC+RHOC1(IR2)*RC1(IP2)*SPC(IP2))
C2158         ZZ=Z8E(JZ2)-ZC(IC)
C2160         RHO2=XX*XX+YY*YY+ZZ*ZZ
C2166         RR=SQRT(RHO2)
C2168         A3=AIC*AJB*CEXP(-U*BK*RR)/4./PI/RR
C2170C CAP-EDGE(BOT) IMPEDANCE ELEMENTS COMPUTED.
C2172C Z(L3) = ZTZ , Z(L4) = ZRZ.
C2174         Z3E=Z3E+TP(IP7)*TCT(IR7)/RHOC1(IR2)*TZ(JP7)*TPBE(JZ7)*A3
C2176         Z4E=Z4E+TZ(IP7)*((TPCR(IR7)/RC1(IP2)+TCR(IR7)/RI)*
C2178       1 TZ(JP7)*TPBE(JZ7)*A3
C2180 171   CONTINUE
C2182 170   CONTINUE
C2202         Z(L3)=U*ETA*(BK*Z3-Z3E/BK)
C2212         Z(L4)=U*ETA*(BK*Z4-Z4E/BK)
C2222 131   CONTINUE
C2232 130   CONTINUE
C2234C EDGE-EDGE LOOP.
C2242         DO 230 JC=1,NC
C2252         DO 230 JP=1,NM
C2272         L4=(LC2+LE)*LC2+((JC-1)*NM+(JP-1))*(LC2+LE)+LC2
C2312         DO 231 IC=1,NC
C2322         DO 231 IP=1,NM
C2332         L4=L4+1
C2342         Z4E=C.0
C2422         JP2=2*(JP-1)
C2432         JP7=4*(JP-1)
C2442         DO 270 JJP=1,4
C2452         JP2=JP2+1
C2462         JP7=JP7+1
C2472         JR2=2*LR
C2482         JR7=2*(JC-1)
C2484         JZ2=2*(JC-1)
C2486         JZ7=2*(JC-1)
C2492         DO 270 JJRZ=1,2
C2502         JR2=JR2+1
C2512         JR7=JR7+1
C2514         JZ2=JZ2+1
C2516         JZ7=JZ7+1
C2522         IP2=2*(IP-1)
C2532         IP7=4*(IP-1)
C2542         DO 271 IIP=1,4
C2552         IP2=IP2+1
C2562         IP7=IP7+1
C2572         IR2=2*LR
C2582         IR7=2*(IC-1)
```

```
C2584         IZZ=2*(IC-1)
C2586         IZ7=2*(IC-1)
C2592         DO 271 IIR2=1,2
C2602         IR2=IR2+1
C2612         IR7=IR7+1
C2613         IZZ=IZZ+1
C2614         IZ7=IZ7+1
C2615         AJC=AC(JP2)*ABS(RHOC(JR2+1)**2-RHOC(JR2)**2)
C2616         AIC=AC(IP2)*ABS(RHOC(IR2+1)**2-RHOC(IR2)**2)
C2617         AJB=DH(JP2)*ABS(ZE(JC)-ZC(JC))/2.
C2618         AIB=DH(IP2)*ABS(ZE(IC)-ZC(IC))/2.
C2622         IF(IC.NE.JC) GO TO 260
C2632         IF(IR2.NE.JR2) GO TO 260
C2642         IF(YB1(IP2).NE.YB1(JP2)) GO TO 260
C2652         IF(XB1(IP2).NE.XB1(JP2)) GO TO 260
C2662C SELF TERM.
C2672         A3=GS(IR2+(IP2-1)*NPR1)
C2682         GO TO 265
C2692C NON-SELF TERM.
C2702  260    CONTINUE
C2712         XX=RHOC1(JR2)*RC1(JP2)*CPC(JP2)-RHOC1(IR2)*RC1(IP2)*CPC(IP2)
C2722         YY=RHOC1(JR2)*RC1(JP2)*SPC(JP2)-RHOC1(IR2)*RC1(IP2)*SPC(I*2)
C2732         ZZ=ZC(JC)-ZC(IC)
C2742         RHO2=XX*XX+YY*YY+ZZ*ZZ
C2772         RR=SQRT(RHO2)
C2782         A3=AIC*AJC*CEXP(-U*BK*RR)/4./PI/RR
C2792  265    CONTINUE
C2802         RI=RHOC1(IR2)*RC1(IP2)
C2812         RJ=RHOC1(JR2)*RC1(JP2)
C2814C EDGE(CAP)-EDGE(CAP) IMPEDANCE ELEMENTS COMPUTED.
C2816         Z4E=Z4E+TZ(IP7)*TZ(JP7)*((CPC(IP2)*CPC(JP2)+
C2818        1 SPC(IP2)*SPC(JP2))*TCE(IR7)*TCE(JR7)-
C2820        2 (TPCE(IR7)/RC1(IP2)+TCE(IR7)/RI)*
C2822        3 (TPCE(JR7)/RC1(JP2)+TCE(JR7)/RJ)/BK/BK)*A3
C2824         XX=XB1(JP2)-(XC+RHOC1(IR2)*RC1(IP2)*CPC(IP2))
C2826         YY=YB1(JP2)-(YC+RHOC1(IR2)*RC1(IP2)*SPC(IP2))
C2828         ZZ=ZBE(JZZ)-ZC(IC)
C2830         RHO2=XX*XX+YY*YY+ZZ*ZZ
C2832         RR=SQRT(RHO2)
C2834         A3=AIC*AJB*CEXP(-U*BK*RR)/4./PI/RR
C2836C EDGE(CAP)-EDGE(BOT) IMPEDANCE ELEMENTS COMPUTED.
C2838         Z4E=Z4E-TZ(IP7)*(TPCE(IP7)/RC1(IP2)+TCE(IR7)/RI)*
C2840        1 TZ(JP7)*TPBE(JZ7)/BK/BK*A3
C2842         XX=XB1(IP2)-(XC+RHOC1(JR2)*RC1(JP2)*CPC(JP2))
C2844         YY=YB1(IP2)-(YC+RHOC1(JR2)*RC1(JP2)*SPC(JP2))
C2846         ZZ=ZBE(IZZ)-ZC(JC)
C2848         RHO2=XX*XX+YY*YY+ZZ*ZZ
C2850         RR=SQRT(RHO2)
C2852         A3=AIB*AJC*CEXP(-U*BK*RR)/4./PI/RR
C2854C EDGE(BOT)-EDGE(CAP) IMPEDANCE ELEMENTS COMPUTED.
C2856         Z4E=Z4E-TZ(IP7)*TPBE(IZ7)*TZ(JP7)*(TPCE(JR7)/RC1(JP2)+
C2858        1 TCE(JR7)/RJ)/BK/BK*A3
```

```
C2860       XX=XB1(JP2)-XB1(IP2)
C2862       YY=YB1(JP2)-YB1(IP2)
C2864       ZZ=ZBE(JZ2)-ZBE(IZ2)
C2866       RHO2=XX*XX+YY*YY+ZZ*ZZ
C2868C CHECK FOR SELF-TERM.
C2870       IF(RHO2.EQ.C.0) RHO2=DH(IP2)*DH(IP2)/4.
C2872       RR=SQRT(RHO2)
C2874       A3=AJB*AJB*CEXP(-U*BK*RR)/4./PI/RR
C2875C EDGE(BOT)-EDGE(BOT) IMPEDANCE ELEMENTS COMPUTED.
C2876       Z4E=Z4E+TZ(IP7)*TZ(JP7)*(TBE(IZ7)*TBE(JZ7)-
C2878      1 TPBE(IZ7)*TPBE(JZ7)/BK/BK)*A3
C2992  271  CONTINUE
C3002  270  CONTINUE
C3042       Z(L4)=U*BK*ETA*Z4E
C3052  231  CONTINUE
C3062  230  CONTINUE
C3063C USE SYMMETRY TO FILL EDGE-CAP MATRIX, FROM CAP-EDGE MATRIX.
C3064       K1=LC2+1
C3065       K2=LC2+LE
C3066       DO 300 J=K1,K2
C3067       DO 300 I=1,LC2
C3068       L=(J-1)*K2+I
C3069       LT=(I-1)*K2+J
C3070       Z(LT)=Z(L)
C3071  300  CONTINUE
C3072C **** END CALCULATION OF ZCC MATRIX ****
C3080  900  CONTINUE
C3090C PRINT ZCC MATRIX BY SUBMATRIX BY COLUMNS.
C3100       LI(1)=LC
C3110       LI(2)=LC
C3120       LI(3)=LE
C3130       LJ(1)=LC
C3140       LJ(2)=LC
C3150       LJ(3)=LE
C3160       CALL ZLIST(3,3,LI,LJ,NAMI,NAMJ,Z)
C3302C WRITE ZCC TO DISK.
C3312       WRITE(7) (Z(I),I=1,LCC)
C3322       STOP
C3332       END
C3340C ---------------------------------------------------------
C3345       SUBROUTINE BOTIN
C3350C
C3355C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
C3360C
C3365       COMMON /BOT1/ NMODE,NPT,NBAND
C3370       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C3375       COMMON /BOT3/ DH(82),SV(82),CV(82)
C3380       COMMON /BOT5/ T(160),TP(160),TZ(160)
C3385       COMMON /BOT6/ IEDGE,IUNIF
C3390       READ(5,49) NMODE,NPT,NBAND
C3395       READ(5,49) NP
C3400  49   FORMAT(3I3)
```

```
      WRITE(6,48) NMODE,NPT,NBAND,NP
   48 FORMAT(32H      NMODE    NPT     NBAND      NP,/,4I8)
      READ(5,53)(YB(I),I=1,NP)
      READ(5,53)(XB(I),I=1,NP)
   53 FORMAT(10F8.4)
      WRITE(6,55)
   55 FORMAT(/,3H YB)
      WRITE(6,46)(YB(I),I=1,NP)
   46 FORMAT(1X,10F8.4)
      WRITE(6,56)
   56 FORMAT(/,3H XB)
      WRITE(6,46)(XB(I),I=1,NP)
C PLOT THE BODY COORDINATES.
      CALL PLOTB(XB,YB,NP,41)
      READ(5,53) BL
      WRITE(6,47) BL
   47 FORMAT(//,21H HALF-LENGTH OF BOT =,F12.4)
C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
      IF(MOD(NP,2).NE.1) GO TO 980
C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
      IEDGE=1
      IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
      IEDGE=0
      YB(NP+1)=YB(2)
      XB(NP+1)=XB(2)
      YB(NP+2)=YB(3)
      XB(NP+2)=XB(3)
      NP=NP+2
      WRITE(6,66) NP
   66 FORMAT(//,39H BOT GENERATING CURVE IS CLOSED.  NP = ,I3)
   10 CONTINUE
C COMPUTATION OF BODY SEGMENT PARAMETERS.
      DO 57 I=2,NP
      I2=I-1
      RR1=YB(I)-YB(I2)
      RR2=XB(I)-XB(I2)
      DH(I2)=SQRT(RR1*RR1+RR2*RR2)
      XB1(I2)=.5*(XB(I)+XB(I2))
      YB1(I2)=.5*(YB(I)+YB(I2))
      SV(I2)=RR1/DH(I2)
      CV(I2)=RR2/DH(I2)
   57 CONTINUE
C CHECK IF BOT SEGMENTATION IS UNIFORM.
      IUNIF=0
      NP1=NP-1
      DO 60 I=2,NP1
      RR1=DH(I)/DH(1)
      IF(RR1.LT.0.99 .OR. RR1.GT.1.01) GO TO 20
   60 CONTINUE
      IUNIF=1
      WRITE(6,67)
   67 FORMAT(//,' BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
```

```
03665 20    CONTINUE
03670C COMPUTATION OF TRIANGLE FUNCTIONS T.
03675       NM=(NP-3)/2
03680       DO 74 J=1,NM
03685       J2=2*(J-1)+1
03690       J3=J2+1
03695       J4=J3+1
03700       J5=J4+1
03705       J6=4*(J-1)+1
03710       J7=J6+1
03715       J8=J7+1
03720       J9=J8+1
03725       DEL1=DH(J2)+DH(J3)
03730       DEL2=DH(J4)+DH(J5)
03735       TP(J6)=1./DEL1
03740       TP(J7)=1./DEL1
03745       TP(J8)=-1./DEL2
03750       TP(J9)=-1./DEL2
03755       T(J6)=DH(J2)/2./DEL1
03760       T(J7)=(DH(J2)+DH(J3)/2.)/DEL1
03765       T(J8)=(DH(J4)/2.+DH(J5))/DEL2
03770       T(J9)=DH(J5)/2./DEL2
03775    74 CONTINUE
03780       NM4=NM*4
03785       DO 75 J=1,NM4
03790 75    TZ(J)=T(J)
03795       IF(IEDGE.EQ.0) GO TO 76
03800       TZ(1)=2.-T(1)
03805       TZ(2)=2.-T(2)
03810       TZ(NM4-1)=2.-T(NM4-1)
03815       TZ(NM4)=2.-T(NM4)
03820 76    CONTINUE
03825       RETURN
03830 980   WRITE(6,981)
03835 981   FORMAT(//,' **** ERROR IN BOT INPUT')
03840       STOP
03845       END
03850C ************************************************************
03855       SUBROUTINE PLOTA(X,Y,N,NP)
03860C ************************************************************
03865C
03870C WRITTEN 2/14/74   BY J. M. PUTNAM       DEPT 220     X23877
03875C
03880C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
03885C    N IS THE NUMBER OF POINTS TO BE PLOTTED.
03890C    NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
03895C    NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
03900C       NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
03905C
03910       REAL X(1),Y(1),HEAD(10)
03915       INTEGER  LINE(101),BLANK,STAR,PLUS
03920       DATA BLANK,STAR,PLUS /1H ,1H*,1H+/
```

```
C3925        NC=51
C3930        N10=(NC-1)/10
C3935        WRITE(6,500)
C3940  500   FORMAT(//,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
C3945        WRITE(6,504)
C3950        XMIN=X(1)
C3955        XMAX=X(1)
C3960        YMIN=Y(1)
C3965        YMAX=Y(1)
C3970        DO 6 I=1,N
C3975        IF(X(I).LT.XMIN) XMIN=X(I)
C3980        IF(X(I).GT.XMAX) XMAX=X(I)
C3985        IF(Y(I).LT.YMIN) YMIN=Y(I)
C3990        IF(Y(I).GT.YMAX) YMAX=Y(I)
C3995  6     CONTINUE
C4000        DEL=XMAX-XMIN
C4005        IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C4010        XMAX=XMIN+DEL
C4015        YMAX=YMIN+DEL
C4020        DO 5 I =1,N10
C4025        Z=I
C4030     5  HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
C4035        DY=(YMAX-YMIN)/(NR-1)
C4040        IEDGE=1
C4045        IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
C4050        Z=YMAX+DY
C4055        YL=Z-DY/2.
C4060        DO 7 J=1,NR
C4065        DO 8 K=1,NC
C4070  8     LINE(K)=BLANK
C4075        Z=Z-DY
C4080        YU=YL
C4085        YL=Z-DY/2.
C4090        DO 9 I=1,N
C4095        IF(Y(I).GE.YU) GO TO 9
C4100        IF(Y(I).LT.YL) GO TO 9
C4105        K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
C4110        IF(K.GT.NC) K=NC
C4115        LINE(K)=STAR
C4120        IF(MOD(I,2).EQ.1) LINE(K)=PLUS
C4125        IF(IEDGE.EQ.0) GO TO 9
C4130        IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
C4135  9     CONTINUE
C4140        WRITE(6,508) Z,(LINE(K),K=1,NC)
C4145  7     CONTINUE
C4150        WRITE(6,504)
C4155        WRITE( 6,3002)
C4160        WRITE( 6,507) XMIN,(HEAD(I),I=1,N10)
C4165        RETURN
C4170  504   FORMAT ( 1X, 14(1H-), 1H., 10(5H----.), 1H- )
C4175  507   FORMAT(10X,11(F10.4))
C4180  508   FORMAT (1X, F12.4,1X, 1HI, 51A1, 1HI )
```

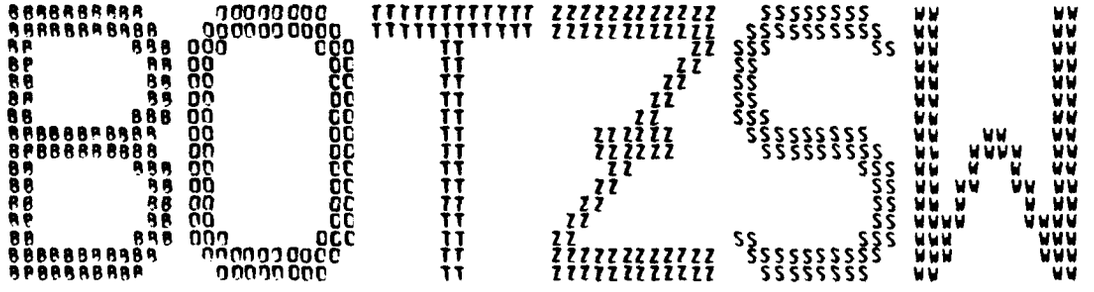
```
04185 3002 FORMAT(4X,7HYH / XH,4X,1HI,5(9X,1HI))
04190      END
04195C -----------------------------------------------------
04200      SUBROUTINE CAPIN
04205C
04210C READ CAP INPUTS AND COMPUTE CAP ARRAYS.
04215C
04220      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
04225      COMMON /CAP1/ NC,XC,YC,ZC(2)
04230      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
04235      COMMON /CAP3/ TCP(36),TCT(36),TPCR(36),TPCT(36)
04240      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
04245      COMMON /EDG1/ NE,ZE(2),ZBE(10)
04250      COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
04255      READ(5,1) NC,NPR,NE
04260 1    FORMAT(3I3)
04265      IF(NE.NE.0) NE=NC
04270      WRITE(6,3) NC,NPR,NE
04275 3    FORMAT(16H1    NC   NPR    NE,//,3I5,//)
04280      IF(NC.EQ.0) RETURN
04285      READ(5,2) XC,YC
04290 2    FORMAT(1CF8.4)
04295      READ(5,2) (ZC(I),I=1,NC)
04300      READ(5,2) (RHOC(I),I=1,NPR)
04305      IF(NE.NE.0) READ(5,2) (ZE(I),I=1,NE)
04310      WRITE(6,4)
04315 4    FORMAT(37H  CAP       XC       YC       ZC       ZE,/)
04320      DO 100 I=1,NC
04325      IF(NE.EQ.0) WRITE(6,5) I,XC,YC,ZC(I)
04330      IF(NE.NE.0) WRITE(6,5) I,XC,YC,ZC(I),ZE(I)
04335 100  CONTINUE
04340 5    FORMAT(I4,4X,4F8.4)
04345      WRITE(6,6)
04350 6    FORMAT(//,5H RHOC)
04355      WRITE(6,7) (RHOC(I),I=1,NPR)
04360 7    FORMAT(1X,10F8.4)
04365      IF(MOD(NPR,2).NE.1) GO TO 980
04370      DO 120 I=2,NPR
04375 120  IF(RHOC(I).LE.RHOC(I-1)) GO TO 980
04380C COMPUTATION OF CAP SECTOR PARAMETERS.
04385      DO 47 I=1,NP
04390      RC(I)=SQRT((YB(I)-YC)**2+(XB(I)-XC)**2)
04395 47   CONTINUE
04400      DO 57 I=2,NP
04405      I2=I-1
04410      RR1=YB1(I2)-YC
04415      RR2=XB1(I2)-XC
04420      RC1(I2)=SQRT(RR1*RR1+RR2*RR2)
04425      AC(I2)=ABS((XB(I)-XC)*(YB(I)+YC)+(XB(I2)-XB(I))*
04430     1  (YB(I2)+YB(I))+(XC-XB(I2))*(YC+YB(I2)))/2.
04435      SPC(I2)=RR1/RC1(I2)
04440      CPC(I2)=RR2/RC1(I2)
```

```
      57    CONTINUE
            DO 67 I=2,NPR
            I2=I-1
            RHOC1(I2)=(RHOC(I)+RHOC(I2))/2.
            DRHOC(I2)=RHOC(I)-RHOC(I2)
      67    CONTINUE
C     COMPUTATION OF CAP TRIANGLE FUNCTIONS.
            LC=(NPR-3)/2
            DO 74 J=1,LC
            J2=2+(J-1)+1
            J3=J2+1
            J4=J3+1
            J5=J4+1
            J6=4+(J-1)+1
            J7=J6+1
            J8=J7+1
            J9=J8+1
            DEL1=DRHOC(J2)+DRHOC(J3)
            DEL2=DRHOC(J4)+DRHOC(J5)
            TPCR(J6)=1./DEL1
            TPCR(J7)=1./DEL1
            TPCR(J8)=-1./DEL2
            TPCR(J9)=-1./DEL2
            TCR(J6)=DRHOC(J2)/2./DEL1
            TCR(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
            TCR(J8)=(DRHOC(J4)/2.+DRHOC(J5))/DEL2
            TCR(J9)=DRHOC(J5)/2./DEL2
      74    CONTINUE
            LC4=LC*4
            DO 75 I=1,LC4
            TCT(I)=TCR(I)
            TPCT(I)=TPCR(I)
      75    CONTINUE
            TCT(LC4-1)=2.-TCT(LC4-1)
            TCT(LC4)=2.-TCT(LC4)
            TPCT(LC4-1)=-TPCT(LC4-1)
            TPCT(LC4)=-TPCT(LC4)
            IF(RHOC(1).EQ.0.0) GO TO 76
            TCT(1)=2.-TCT(1)
            TCT(2)=2.-TCT(2)
            TPCT(1)=-TPCT(1)
            TPCT(2)=-TPCT(2)
      76    CONTINUE
            IF(NE.EQ.0) RETURN
C     COMPUTATION OF EDGE HALF TRIANGLE FUNCTIONS.
            DO 80 IC=1,NC
            J2=NPP-2
            J3=J2+1
            J6=2+(IC-1)+1
            J7=J6+1
            DEL1=DRHOC(J2)+DRHOC(J3)
            TPCE(J6)=1./DEL1
```

```
C4705       TPCE(J7)=1./DEL1
C4710       TCE(J6)=DRHOC(J2)/2./DEL1
C4715       TCE(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
C4720       DEL2=ZE(IC)-ZC(IC)
C4725       ZBE(J6)=ZC(IC)+0.25*DEL2
C4730       ZBE(J7)=ZC(IC)+0.75*DEL2
C4735       IF(DEL2.LT.0.0) GO TO 78
C4740C EDGE IS AT Z=-L.
C4745       TPBE(J6)=-1./DEL2
C4750       TPBE(J7)=-1./DEL2
C4755       TBE(J6)=0.75
C4760       TBE(J7)=0.25
C4765       GO TO 80
C4770 78    CONTINUE
C4775C EDGE IS AT Z=+L.
C4780       TPBE(J6)=1./DEL2
C4785       TPBE(J7)=1./DEL2
C4790       TBE(J6)=-0.75
C4795       TBE(J7)=-0.25
C4800 80    CONTINUE
C4805       RETURN
C4810 980   WRITE(6,981)
C4815 981   FORMAT(///,' **** ERROR IN CAP INPUT')
C4820       STOP
C4825       END
C6212C ******************************************************
C6222       SUBROUTINE CSIMP(F,A,B,DEL,IMAX,SI1,S,N,IER)
C6232C ******************************************************
C6242C COMPLEX SIMPSON INTEGRATION ROUTINE.
C6252       COMPLEX F,SI1,S,SUMK
C6262       SI1=(.0,.0)
C6272       S=(.C,.C)
C6282       N=0
C6292       BA=B-A
C6302       IF (BA)20,19,20
C6312    19 IER=1
C6322       RETURN
C6332    20 IF(DEL)22,22,23
C6342    22 IER=2
C6352       RETURN
C6362    23 IF(IMAX-1)24,24,25
C6372    24 IER=3
C6382       RETURN
C6392    25 X=BA/2.+A
C6402       NHALF=1
C6412       SUMK=F(X)*BA*2./3.
C6422       S=SUMK+(F(A)+F(B))*BA/6.
C6432       DO 28 I=2,IMAX
C6442       SI1=S
C6452       S=(S-SUMK/2.)/2.
C6462       NHALF=NHALF*2
C6472       ANHLF=NHALF
```

```
      FRSTX=A+(BA/ANHLF)/2.
      SUMK=F(FRSTX)
      XK=FRSTX
      KLAST=NHALF-1
      FINC=BA/ANHLF
      DO 26 K=1,KLAST
      XK=XK+FINC
   26 SUMK=SUMK+F(XK)
      SUMK=SUMK*2.*BA/(3.*ANHLF)
      S=S+SUMK
      IF(CABS(S).EQ.0.0) GO TO 29
      IF((CABS(S-SI1)/CABS(S))-DEL) 29,28,28
   28 CONTINUE
      IER=4
      WRITE(6,1) SII,S
    1 FORMAT(///,29H INTEGRATION DID NOT CONVERGE,//
     1 30H THE PREVIOUS AND FINAL VALUES OF THE INTEGRAL ARE,
     2 13H RESPECTIVELY,4E12.3)
      GO TO 30
   29 IER=0
   30 N=2*NHALF
      RETURN
      END
C-----------------------------------------------
      COMPLEX FUNCTION FUNC(R)
      COMPLEX U
      COMMON /WAVE/ BK
      COMMON /INT/ CONST1,CONST2
      DATA U /(0.,1.)/
      T=SQRT(CONST1+R*R+R*CONST2)
      FUNC=R*CEXP(-U*BK*T)/T
      RETURN
      END
C-----------------------------------------------
      SUBROUTINE ZLIST(NI,NJ,LI,LJ,NAMI,NAMJ,Z)
C WRITE Z TO THE LINE-PRINTER BY SUBMATRIX BY COLUMNS.
      COMPLEX Z(1)
      DIMENSION LI(1),LJ(1),NAMI(1),NAMJ(1)
      NCOL=0
      DO 1CC I=1,NI
  1CC NCOL=NCOL+LI(I)
      NR=1
      DO 3CC J=1,NJ
      IF(LJ(J).EQ.0) GO TO 300
      NC=1
      DO 2CC I=1,NI
      IF(LI(I).EQ.0) GO TO 200
      WRITE(6,1) NAMI(I),NAMJ(J)
    1 FORMAT(/,2H Z,1X,A4,3H - ,A4)
      KI=LI(I)
      KJ=LJ(J)
      K1=(NR-1)*NCOL+NC
```

```
C7000         DO 150 KK=1,KJ
C7010         K2=K1+KI-1
C7020         WRITE(6,2) (Z(K),K=K1,K2)
C7030    2    FORMAT(1X,10G11.4)
C7040  150    K1=K1+NCOL
C7050         NC=NC+LI(I)
C7060  200    CONTINUE
C7070         NR=NR+LJ(J)
C7080  300    CONTINUE
C7090         RETURN
C7100         END
07110/EOR
```



```
CC100BOTZ,P3C,T100,CM140000,STCX3.
CC110ACCOUNT,M0326S,BOT2A.
CC120BANNERS(OUTPUT) *J. PUTNAM*DEPT 220*BLD 110-4***
CC130DEFINE(OUTFIL=ZSWX)
CC138FTN(R=0,OPT=2)
CC140LDSET(PRESET=INDEF)
CC150LGO.
CC210EXIT.
CC230/EOR
CC240       PROGRAM BOTZSW(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
CC250      1 OUTFIL,TAPE7=OUTFIL)
CC252C
CC254C BOTZSW GENERATES THE BOT-WIRE/JUNCTION MATRIX.
CC256C
CC258C UNIT 5 IS THE CARD READER.
CC260C UNIT 6 IS THE LINE PRINTER.
CC262C UNIT 7 IS A DISK FILE FOR OUTPUT OF THE Z MATRIX.
CC264C
CC280       COMPLEX A3,Z(6400),G1(3403),G2(3403),U
CC281       COMPLEX G1J,G2J
CC282       COMMON /WAVE/ BK
CC284       COMMON /BOT1/ NMODE,NPT,NBAND
CC286       COMMON /BOT2/ NP,RL,YB(83),XB(83),YB1(82),XB1(82)
CC288       COMMON /BOT3/ DH(82),SV(82),CV(82)
CC290       COMMON /BOT5/ T(160),TP(160),TZ(160)
CC291       COMMON /BOT6/ IEDGE,IUNIF
CC292       COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
CC293      1 XW1(100),YW1(100),ZW1(100)
CC294       COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
CC295       COMMON /WIRE3/ NW,INDW(6),RADW(5)
CC296       COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
CC297       COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
CC298       COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
CC302       COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
CC304       COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
CC306       COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
CC308       COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
CC320       COMMON /INT/ M,RHO2,ZP
CC325       DIMENSION LI(2),LJ(2),NAMI(2),NAMJ(2)
CC330       EXTERNAL FUNC1,FUNC2
CC340       DATA NAMI,NAMJ /4HS(I),4HS(Z),4HWIRE,4HJUNC/
CC430       U=(0.,1.)
CC432       PI=3.14159265
CC434       ETA=376.707
CC435       READ(5,1) BK
CC436 1     FORMAT(E15.7)
CC437       WRITE(6,2) BK
CC438 2     FORMAT(9H1       BK=,/,E15.7)
CC440C
CC450       CALL BOTIN
CC460C
CC470       KG=NP-1
```

```
C0480       NM=(NP-3)/2
C0490       NM2=NM*2
C0500       BK2=BK*BK
C0510       BKL=BK*BL
00790C
C0800       CALL SCAPIN
00835C
C0836       CALL WIREIN
00837C
C0838       WRITE(7) NP,NW,NPW,NJ
C0840       KGW=NPW-1
C0842       LSW=NM2*LW
C0844       LSWJ=NM2*(LW+NJ)
00860C CONSTANTS USED IN SUMMATION OF IMPEDANCE ELEMENTS.
C0865       C1=BK*ETA
C0870       C2=C1
00890C *** END INITIALIZATION OF PARAMETERS ***
00939C COMPUTATION OF IMPEDANCE MATRICES Z(M), WHERE M = -NMODE+1 TO
C0941C NMODE-1.
C0943       M=-NMODE
C0947       KMODE=2*NMODE-1
C0962       DO 998 MM=1,KMODE
C0963       M=M+1
C0965       WRITE(6,54) M
C0970 54    FORMAT(4H1 M=,I3)
C1000C COMPUTATION OF GREEN'S FUNCTION KERNEL.
C1010       I1=0
C1020       NPT1=0
C1030       NINT1=0
C1040       NPT2=0
C1050       NINT2=0
C1188       IW=1
C1190       DO 16 J=1,KGW
C1192C DETERMINE IF SEGMENT J CONNECTS 2 DISTINCT WIRES.  IF SO, SKIP SEGMENT J.
C1200       IF(J+1.EQ.INDW(IW)) GO TO 7
C1230       DO 17 I=1,KG
C1240       I1=I1+1
C1260       YY=YB1(I)-YW1(J)
C1270       XX=XB1(I)-XW1(J)
C1370       RHO2=XX*XX+YY*YY
C1380       ZP=ZW1(J)
C1400       CALL CSIMP(FUNC2,-BL,BL,0.05,10,A3,G2(I1),K,IER)
C1402       NPT2=NPT2+K+1
C1404       NINT2=NINT2+1
C1405       G2(I1)=G2(I1)/4./PI
C1410       IF(MP.NE.1) GO TO 17
C1420       CALL CSIMP(FUNC1,-BL,BL,0.05,10,A3,G1(I1),K,IER)
C1430       NPT1=NPT1+K+1
C1440       NINT1=NINT1+1
C1450       G1(I1)=G1(I1)/4./PI
C1650    17 CONTINUE
C1652       GO TO 16
```

```
01654  7      I1=I1+KG
01656         IW=IW+1
01660    16 CONTINUE
01661         AVG1=0.0
01662         IF(MP.EQ.1) AVG1=NPT1/FLOAT(NINT1)
01664         AVG2=NPT2/FLOAT(NINT2)
01666         WRITE(6,56) NINT1,AVG1,NINT2,AVG2
01668  56     FORMAT(/,' NUMBER OF INTEGRATIONS AND AVERAGE NUMBER OF',
01670        1 ' POINTS PER INTEGRATION (G1 AND G2, RESPECTIVELY)',/,
01672        2 2(I8,G1C.3))
01690C *** END CALCULATION OF GREEN'S KERNEL ***
C2005C COMPUTATION OF IMPEDANCE ELEMENTS.
C2007C BOT-WIRE LOOP.
02010         DO 30 J=1,LW
02020         JL=(J-1)*NM2
02030         J3=(J-1)*4
02040         JJ=INDTW(J)-1
02050         DO 31 I=1,NM
02055C COMPUTE SUBMATRIX INDICES.
02060         L1=JL+I
02070         L2=L1+NM
02160         Z(L1)=0.
02110         Z(L2)=0.
02150         II=2*(I-1)
02160         I3=(I-1)*4
02170         DO 70 JJ=1,4
02180         J2=JJ+JJ
02190         J7=J3+JJ
02195         J4=(J2-1)*KG+I1
02200         DO 71 II=1,4
02210         I2=I1+II
02220         I7=I3+II
02230         J4=J4+1
02295C BOT-WIRE IMPEDANCE ELEMENTS COMPUTED.
02297C Z(L1) = ZT , Z(L2) = ZZ.
02300         Z(L1)=Z(L1)+DH(I2)*((CV(I2)*DXW(J2)+SV(I2)*DYW(J2))*
02310        1 T(I7)*TW(J7)-DHW(J2)*TP(I7)*TPW(J7)/BK2)*G2(J4)
02320         Z(L2)=Z(L2)+DH(I2)*TZ(I7)*(DZW(J2)*TW(J7)*(G2(J4)-
02330        1 (-1)**M*G1(J4))+U*M*PI*DHW(J2)*TPW(J7)/BK/BKL*G2(J4))
02370     71 CONTINUE
02375     70 CONTINUE
02381         Z(L1)=U*C1*Z(L1)
02382         Z(L2)=U*C2*Z(L2)
02406  31     CONTINUE
C2407  30     CONTINUE
02410         IF(NJ.EQ.0) GO TO 900
02417C BOT-JUNCTION LOOP.
02427         DO 130 J=1,NJ
C2437         JL=(J-1)*NM2+LSW
02447         J3=(J-1)*2
C2457         JJ=INDTJ(J)-1
C2467         DO 131 I=1,NM
```

```
C2477C COMPUTE SUBMATRIX INDICES.
C2487       L1=JL+I
C2497       L2=L1+NM
C2507       Z(L1)=0.
C2517       Z(L2)=0.
C2527       I1=2*(I-1)
C2537       I3=(I-1)*4
C2547       DO 170 JJ=1,2
C2557       J2=J1+JJ
C2567       J7=J3+JJ
C2577       J4=(J2-1)*KG+I1
C2587       DO 171 II=1,4
C2597       I2=I1+II
C2607       I7=I3+II
C2617       J4=J4+1
C2627C BOT-JUNCTION(WIRE) IMPEDANCE ELEMENTS COMPUTED.
C2629C Z(L1) = ZT , Z(L2) = ZZ.
C2637       Z(L1)=Z(L1)+DH(I2)*((CV(I2)*DXW(J2)+SV(I2)*DYW(J2))*
C2647      1 T(I7)*TJ(J7)-D4W(J2)*TP(I7)*TPJ(J7)/BK2)*G2(J4)
C2657       Z(L2)=Z(L2)+DH(I2)*TZ(I7)*(DZW(J2)*TJ(J7)*(G2(J4)-
C2667      1 (-1)**M*G1(J4))+U*M*PI*D4W(J2)*TPJ(J7)/BK/BKL*G2(J4))
C2668 171   CONTINUE
C2670 170   CONTINUE
C2672       DO 270 JJ=1,4
C2674       ALPHA=PI*((JJ-1)/2.+0.25)
C2676       SA=SIN(ALPHA)
C2678       CA=COS(ALPHA)
C2680       RA=(RADJ(J)+RADD(J))/2.
C2682       XA=XJ(J)+RA*(CA*UXJ1(J)+SA*UXJ2(J))
C2684       YA=YJ(J)+RA*(CA*UYJ1(J)+SA*UYJ2(J))
C2686       ZA=ZJ(J)+RA*(CA*UZJ1(J)+SA*UZJ2(J))
C2688       DO 271 II=1,4
C2690       I2=I1+II
C2692       I7=I3+II
C2694       YY=YB1(I2)-YA
C2696       XX=XB1(I2)-XA
C2698       RHO2=XX*XX+YY*YY
C2700       ZP=ZA
C2702       CALL CSIMP(FUNC1,-BL,BL,0.05,10,A3,G1J,K,IER)
C2704       G1J=G1J/4./PI
C2706       CALL CSIMP(FUNC2,-BL,BL,0.05,10,A3,G2J,K,IER)
C2707       G2J=G2J/4./PI
C2708       DT=CA*(CV(I2)*UXJ1(J)+SV(I2)*UYJ1(J))+
C2709      1 SA*(CV(I2)*UXJ2(J)+SV(I2)*UYJ2(J))
C2710       DZ=CA*UZJ1(J)+SA*UZJ2(J)
C2711       RR=(RADJ(J)-RADD(J))/2.
C2712C BOT-JUNCTION(DISK) IMPEDANCE ELEMENTS COMPUTED.
C2713C Z(L1) = ZT , Z(L2) = ZZ.
C2714       Z(L1)=Z(L1)+DH(I2)/4.*(DT*T(I7)*RR-
C2715      1 TP(I7)/BK2)*G2J
C2716       Z(L2)=Z(L2)+DH(I2)*TZ(I7)/4.*(DZ*RR*(G2J-(-1)**M*G1J)+
C2717      1 U*M*PI/BK/BKL*G2J)
```

```
 271  CONTINUE
 270  CONTINUE
      Z(L1)=U*C1*Z(L1)
      Z(L2)=U*C2*Z(L2)
 131  CONTINUE
 130  CONTINUE
C*** END CALCULATION OF Z(M) SUBMATRIX ***
 900  CONTINUE
C PRINT OUT IMPEDANCE MATRIX Z(M) BY SUBMATRIX BY COLUMNS.
      LI(1)=NM
      LI(2)=NM
      LJ(1)=LW
      LJ(2)=NJ
      CALL ZLIST(2,2,LI,LJ,NAMI,NAMJ,Z)
C WRITE Z(M) TO DISK.
      WRITE(7) M
      WRITE(7) (Z(I),I=1,LSWJ)
 998  CONTINUE
      STOP
      END
C-----------------------------------------------------------------
      SUBROUTINE BOTIN
C
C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
C
      COMMON /BOT1/ NMODE,NPT,NBAND
      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
      COMMON /BOT3/ DH(82),SV(82),CV(82)
      COMMON /BOT5/ T(160),TP(160),TZ(160)
      COMMON /BOT6/ IEDGE,IUNIF
      READ(5,49) NMODE,NPT,NBAND
      READ(5,49) NP
  49  FORMAT(3I3)
      WRITE(6,48) NMODE,NPT,NBAND,NP
  48  FORMAT(32H      NMODE    NPT     NBAND      NP,/,4I8)
      READ(5,53)(YB(I),I=1,NP)
      READ(5,53)(XB(I),I=1,NP)
  53  FORMAT(10F8.4)
      WRITE(6,55)
  55  FORMAT(//,3H YB)
      WRITE(6,46)(YB(I),I=1,NP)
  46  FORMAT(1X,10F8.4)
      WRITE(6,56)
  56  FORMAT(//,3H XB)
      WRITE(6,46)(XB(I),I=1,NP)
C PLOT THE BODY COORDINATES.
      CALL PLOTB(XB,YB,NP,41)
      READ(5,53) BL
      WRITE(6,47) BL
  47  FORMAT(//,21H HALF-LENGTH OF BOT =,F12.4)
C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
      IF(MOD(NP,2).NE.1) GO TO 980
```

```
C3110C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
C3115        IEDGE=1
C3120        IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
C3125        IEDGE=0
C3130        YB(NP+1)=YB(2)
C3135        XB(NP+1)=XB(2)
C3140        YB(NP+2)=YB(3)
C3145        XB(NP+2)=XB(3)
C3150        NP=NP+2
C3155        WRITE(6,66) NP
C3160 66     FORMAT(//,39H BOT GENERATING CURVE IS CLOSED.  NP = ,I3)
C3165 10     CONTINUE
C3170C COMPUTATION OF BODY SEGMENT PARAMETERS.
C3175        DO 57 I=2,NP
C3180        I2=I-1
C3185        RR1=YB(I)-YB(I2)
C3190        RR2=XB(I)-XB(I2)
C3195        DH(I2)=SQRT(RR1*RR1+RR2*RR2)
C3200        XB1(I2)=.5*(XB(I)+XB(I2))
C3205        YB1(I2)=.5*(YB(I)+YB(I2))
C3210        SV(I2)=RR1/DH(I2)
C3215        CV(I2)=RR2/DH(I2)
C3220 57     CONTINUE
C3225C CHECK IF BOT SEGMENTATION IS UNIFORM.
C3230        IUNIF=0
C3235        NP1=NP-1
C3240        DO 60 I=2,NP1
C3245        RR1=DH(I)/DH(1)
C3250        IF(RR1.LT.0.99 .OR. RR1.GT.1.01) GO TO 20
C3255 60     CONTINUE
C3260        IUNIF=1
C3265        WRITE(6,67)
C3270 67     FORMAT(//,' BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
C3275 20     CONTINUE
C3280C COMPUTATION OF TRIANGLE FUNCTIONS T.
C3285        NM=(NP-3)/2
C3290        DO 74 J=1,NM
C3295        J2=2*(J-1)+1
C3300        J3=J2+1
C3305        J4=J3+1
C3310        J5=J4+1
C3315        J6=4*(J-1)+1
C3320        J7=J6+1
C3325        J8=J7+1
C3330        J9=J8+1
C3335        DEL1=DH(J2)+DH(J3)
C3340        DEL2=DH(J4)+DH(J5)
C3345        TP(J6)=1./DEL1
C3350        TP(J7)=1./DEL1
C3355        TP(J8)=-1./DEL2
C3360        TP(J9)=-1./DEL2
C3365        T(J6)=DH(J2)/2./DEL1
```

```
C3370          T(J7)=(DH(J2)+DH(J3)/2.)/DEL1
C3375          T(J8)=(DH(J4)/2.+DH(J5))/DEL2
C3380          T(J9)=DH(J5)/2./DEL2
C3385    74    CONTINUE
C3390          NM4=NM*4
C3395          DO 75 J=1,NM4
C3400  75       TZ(J)=T(J)
C3405          IF(IEDGE.EQ.0) GO TO 76
C3410          TZ(1)=2.-T(1)
C3415          TZ(2)=2.-T(2)
C3420          TZ(NM4-1)=2.-T(NM4-1)
C3425          TZ(NM4)=2.-T(NM4)
C3430  76    CONTINUE
C3435          RETURN
C3440  980   WRITE(6,981)
C3445  981   FORMAT(//,' **** ERROR IN BOT INPUT')
C3450          STOP
C3455          END
C3455C ------------------------------------------------------
C3856          SUBROUTINE SCAPIN
C3857C
C3858C SKIP CAP INPUTS.
C3859C
C3860          READ(5,1) NC,NPR,NE
C3861  1       FORMAT(3I3)
C3862          IF(NC.EQ.0) RETURN
C3863          IF(NE.NE.0) NE=NC
C3865          READ(5,2) XC,YC
C3866  2       FORMAT(10F8.4)
C3868          READ(5,2) (ZC,I=1,NC)
C3869          READ(5,2) (PHC,I=1,NPR)
C3870          IF(NE.EQ.0) RETURN
C3871          READ(5,2) (ZE,I=1,NE)
C3872          RETURN
C3873          END
C3875C ------------------------------------------------------
C3877          SUBROUTINE WIREIN
C3887C
C3897C READ WIRE COORDINATES AND COMPUTE WIRE SEGMENT ARRAYS.
C3907C
C3937          COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
C3942         1 XW1(100),YW1(100),ZW1(100)
C3947          COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
C3957          COMMON /WIRE3/ NW,INDW(6),RADW(5)
C3967          COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
C3977          COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
C3987          COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
C3988          COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
C3990          COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
C3991          COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
C3992          COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
C3993          NW=0
```

```
03994       NJ=0
03995       LW=0
03997       READ(5,51) NW,NPW,NJ
04007 51    FORMAT(3I3)
C4008       WRITE(6,1) NW,NPW,NJ
04009 1     FORMAT(24H1      NW      NPW      NJ,//,3I8)
C4010       IF(NW.EQ.0) RETURN
04017       READ(5,53) (XW(I),I=1,NPW)
04027       READ(5,53) (YW(I),I=1,NPW)
04037       READ(5,53) (ZW(I),I=1,NPW)
C4047 53    FORMAT(10F8.4)
C4057       READ(5,52) (INDW(I),I=1,NW)
C4060 52    FORMAT(10I8)
04067       INDW(NW+1)=NPW+1
04070       READ(5,53) (RADW(I),I=1,NW)
C4072       IF(NJ.EQ.0) GO TO 50
C4077C NOTE, INDJW MUST BE MONOTONIC INCREASING.
04078       READ(5,52) (INDJW(I),I=1,NJ)
C4086       READ(5,53) (RADD(I),I=1,NJ)
04082       READ(5,53) (UXJ(I),I=1,NJ)
C4084       READ(5,53) (UYJ(I),I=1,NJ)
04086       READ(5,53) (UZJ(I),I=1,NJ)
04097 50    IERW=0
04107       IERJ=0
04117       WRITE(6,61)
04120 61    FORMAT(/,22X,'WIRE COORDINATES',20X,'JUNCTION PARAMETERS',//,
04122      1 26X,'IW',4X,'XW',6X,'YW',6X,'ZW',
04124      2 10X,'IJ',4X,'RADD',5X,'UXJ',5X,'UYJ',5X,'UZJ')
04137       IJ=1
C4138C THIS LOOP LISTS WIRE/JUNCTION POINTS, WHILE CHECKING THE FOLLOWING:
04139C    1) EACH WIRE MUST CONTAIN AN ODD NUMBER OF POINTS.
04140C    2) EACH JUNCTION MUST EITHER START OR TERMINATE A WIRE.
C4143C    3) CHECK THAT ALL JUNCTION POINTS ARE FOUND.
04147       DO 100 IW=1,NW
04157       WRITE(6,62) IW,RADW(IW)
04167 62    FORMAT(2X,'WIRE',I3,' RADW=',F8.4)
04177       I1=INDW(IW)
04187       I2=INDW(IW+1)-1
04197C CHECK FOR AN ODD NUMBER OF POINTS ON WIRE IW.
04207       IF(MOD(I2-I1+1,2).NE.1) IERW=1
04217       DO 90 I=I1,I2
04227       WRITE(6,63) I,XW(I),YW(I),ZW(I)
04237 63    FORMAT(25X,I3,3F8.4)
C4247C CHECK IF WIRE POINT I IS A JUNCTION POINT.
04257       IF(IJ.GT.NJ) GO TO 90
04267       IF(INDJW(IJ).NE.I) GO TO 90
04270C CHECK THAT JUNCTION POINT IJ IS AT THE START OR END OF WIRE.
04272       IF(I.NE.I1 .AND. I.NE.I2) IERJ=1
04300       WRITE(6,64) IJ,RADD(IJ),UXJ(IJ),UYJ(IJ),UZJ(IJ)
C4310 64    FORMAT('+',58X,I3,4F8.4)
04377C DETERMINE DIRECTION IN WHICH WIRE LEAVES JUNCTION POINT.
04387       INDTJ(IJ)=1
```

```
C4397       IF(I.EQ.I2) INDTJ(IJ)=-1
C4398C COMPUTE JUNCTION PARAMETERS.
C4400       RADJ(IJ)=RADW(IW)
C4405       XJ(IJ)=XW(I)
C4406       YJ(IJ)=YW(I)
C4407       ZJ(IJ)=ZW(I)
C4408       IJ=IJ+1
C4477 90    CONTINUE
C4487 100   CONTINUE
C4497C CHECK FOR WIRE OR JUNCTION INPUT ERRORS.
C4507       IF(IERW.NE.0) GO TO 980
C4517       IF(IJ-1.NE.NJ .OR. IERJ.NE.0) GO TO 990
C4527C COMPUTATION OF WIRE SEGMENT PARAMETERS.
C4537       DO 57 I=2,NPW
C4547       I2=I-1
C4557       DXW(I2)=XW(I)-XW(I2)
C4567       DYW(I2)=YW(I)-YW(I2)
C4577       DZW(I2)=ZW(I)-ZW(I2)
C4587       DHW(I2)=SQRT(DXW(I2)**2+DYW(I2)**2+DZW(I2)**2)
C4588       XW1(I2)=0.5*(XW(I)+XW(I2))
C4590       YW1(I2)=0.5*(YW(I)+YW(I2))
C4592       ZW1(I2)=0.5*(ZW(I)+ZW(I2))
C4597    57 CONTINUE
C4607C COMPUTATION OF WIRE TRIANGLE FUNCTIONS TW.
C4617       LW=0
C4627       DO 75 IW=1,NW
C4637       I1=INDW(IW)
C4647       I2=INDW(IW+1)-1
C4657       LW1=(I2-I1-2)/2
C4667       DO 74 J=1,LW1
C4677       LW=LW+1
C4687       J2=2*(J-1)+I1
C4697       J3=J2+1
C4707       J4=J3+1
C4717       J5=J4+1
C4727       J6=4*(LW-1)+1
C4737       J7=J6+1
C4747       J8=J7+1
C4757       J9=J8+1
C4767       INDTW(LW)=J2
C4777       DEL1=DHW(J2)+DHW(J3)
C4787       DEL2=DHW(J4)+DHW(J5)
C4797       TPW(J6)=1./DEL1
C4807       TPW(J7)=1./DEL1
C4817       TPW(J8)=-1./DEL2
C4827       TPW(J9)=-1./DEL2
C4837       TW(J6)=DHW(J2)/2./DEL1
C4847       TW(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C4857       TW(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C4867       TW(J9)=DHW(J5)/2./DEL2
C4877 74    CONTINUE
C4887 75    CONTINUE
```

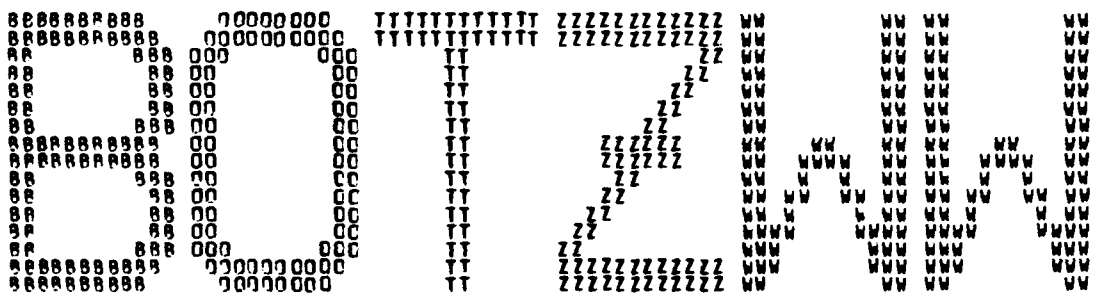
```
C4897C NOTE, LW=(NPW-NW)/2-NW
C4907       IF(NJ.EQ.0) RETURN
C4917C COMPUTATION OF JUNCTION HALF TRIANGLE FUNCTIONS TJ.
C4917       DO 85 IJ=1,NJ
C4927       IF(INDTJ(IJ).GT.0) GO TO 80
C4937C JUNCTION IS AT THE END OF A WIRE.
C4937       J2=INDJW(IJ)-2
C4947       J3=J2+1
C4957       J6=2*(IJ-1)+1
C4967       J7=J6+1
C4977       INDTJ(IJ)=J2
C4987       DEL1=-DHW(J2)-DHW(J3)
C4997       TPJ(J6)=1./DEL1
C5007       TPJ(J7)=1./DEL1
C5017       TJ(J6)=DHW(J2)/2./DEL1
C5027       TJ(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C5037       GO TO 85
C5047C JUNCTION IS AT THE START OF A WIRE.
C5047 80    J4=INDJW(IJ)
C5057       J5=J4+1
C5067       J8=2*(IJ-1)+1
C5077       J9=J8+1
C5087       INDTJ(IJ)=J4
C5097       DEL2=DHW(J4)+DHW(J5)
C5107       TPJ(J8)=-1./DEL2
C5117       TPJ(J9)=-1./DEL2
C5127       TJ(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C5137       TJ(J9)=DHW(J5)/2./DEL2
C5147 85    CONTINUE
C5157C COMPUTE UNIT VECTORS FOR JUNCTION(DISK).
C5167       DO 95 IJ=1,NJ
C5177C NORMAL UNIT VECTOR (UJ).
C5187       RR=SQRT(UXJ(IJ)**2+UYJ(IJ)**2+UZJ(IJ)**2)
C5197       UXJ(IJ)=UXJ(IJ)/RR
C5207       UYJ(IJ)=UYJ(IJ)/RR
C5217       UZJ(IJ)=UZJ(IJ)/RR
C5227C FIND 2 ORTHOGONAL UNIT VECTORS IN THE PLANE OF THE DISK (UJ1 & UJ2).
C5237       UXJ1(IJ)=0.0
C5247       UYJ1(IJ)=0.0
C5257       UZJ1(IJ)=0.0
C5267       UXJ2(IJ)=0.0
C5277       UYJ2(IJ)=0.0
C5287       UZJ2(IJ)=0.0
C5297       IF(UXJ(IJ).EQ.0.0) GO TO 91
C5307C FIND INTERSECTION WITH X-Y PLANE.
C5317       UYJ1(IJ)=1.0
C5327       UXJ1(IJ)=-UYJ(IJ)/UXJ(IJ)
C5337C FIND INTERSECTION WITH X-Z PLANE.
C5347       UZJ2(IJ)=1.0
C5357       UXJ2(IJ)=-UZJ(IJ)/UXJ(IJ)
C5367       GO TO 94
C5377 91    IF(UYJ(IJ).EQ.0.0) GO TO 92
```

```
C5387C FIND INTERSECTION WITH Y-Z PLANE.
C5397       UZJ1(IJ)=1.0
C5407       UYJ1(IJ)=-UZJ(IJ)/UYJ(IJ)
C5417C FIND INTERSECTION WITH X-Y PLANE.
C5427       UXJ2(IJ)=1.0
C5437       UYJ2(IJ)=-UXJ(IJ)/UYJ(IJ)
C5447       GO TO 94
C5457 92    IF(UZJ(IJ).EQ.0.0) GO TO 94
C5467C FIND INTERSECTION WITH X-Z PLANE.
C5477       UXJ1(IJ)=1.0
C5487       UZJ1(IJ)=-UXJ(IJ)/UZJ(IJ)
05497C FIND INTERSECTION WITH Y-Z PLANE.
C5507       UYJ2(IJ)=1.0
C5517       UZJ2(IJ)=-UYJ(IJ)/UZJ(IJ)
C5527 94    CONTINUE
05537       RR=SQRT(UXJ1(IJ)**2+UYJ1(IJ)**2+UZJ1(IJ)**2)
05547       UXJ1(IJ)=UXJ1(IJ)/RR
05557       UYJ1(IJ)=UYJ1(IJ)/RR
C5567       UZJ1(IJ)=UZJ1(IJ)/RR
C5577       RR=SQRT(UXJ2(IJ)**2+UYJ2(IJ)**2+UZJ2(IJ)**2)
C5587       UXJ2(IJ)=UXJ2(IJ)/RR
C5597       UYJ2(IJ)=UYJ2(IJ)/RR
056C7       UZJ2(IJ)=UZJ2(IJ)/RR
C5637 95    CONTINUE
C5647       RETURN
05657 980   WRITE(6,981)
C5667 981   FORMAT(//,' **** ERROR IN WIRE INPUT')
C5677       STOP
05687 990   WRITE(6,991)
C5697 991   FORMAT(//,' **** ERROR IN JUNCTION INPUT')
C57C7       STOP
C5717       END
05727C ************************************************************
05737       SUBROUTINE CSIMP(F,A,B,DEL,IMAX,SI1,S,N,IER)
05747C ************************************************************
C5757C COMPLEX SIMPSON INTEGRATION ROUTINE.
05767       COMPLEX F,SI1,S,SUMK
C5777       SI1=(.0,.0)
05787       S=(.0,.0)
C5797       N=0
C58C7       BA=B-A
05817       IF (BA)20,19,20
05827    19 IER=1
C5837       RETURN
05847    20 IF(DEL)22,22,23
05857    22 IER=2
05867       RETURN
C5877    23 IF(IMAX-1)24,24,25
C5887    24 IER=3
05897       RETURN
C5907    25 X=BA/2.+A
C5917       NHALF=1
```

```
      SUMK=F(X)*BA*2./3.
      S=SUMK+(F(A)+F(B))*BA/6.
      DO 28 I=2,IMAX
      SI1=S
      S=(S-SUMK/2.)/2.
      NHALF=NHALF*2
      ANHLF=NHALF
      FRSTX=A+(BA/ANHLF)/2.
      SUMK=F(FRSTX)
      XK=FRSTX
      KLAST=NHALF-1
      FINC=BA/ANHLF
      DO 26 K=1,KLAST
      XK=XK+FINC
   26 SUMK=SUMK+F(XK)
      SUMK=SUMK*2.*BA/(3.*ANHLF)
      S=S+SUMK
      IF(CABS(S).EQ.0.0) GO TO 29
      IF((CABS(S-SI1)/CABS(S))-DEL) 29,28,28
   28 CONTINUE
      IER=4
      WRITE(6,1) SI1,S
    1 FORMAT(//,29H INTEGRATION DID NOT CONVERGE,//
     1 50H THE PREVIOUS AND FINAL VALUES OF THE INTEGRAL ARE,
     2 13H RESPECTIVELY,4E12.3)
      GO TO 30
   29 IER=0
   30 N=2*NHALF
      RETURN
      END
C*************************************************************
      COMPLEX FUNCTION FUNC1(E)
C*************************************************************
      COMPLEX U
      COMMON /WAVE/ BK
      COMMON /INT/ M,RHO2,ZP
      DATA PI,U /3.14159265,(0.,1.)/
      R=SQRT(RHO2+(E-ZP)**2)
      FUNC1=CEXP(-U*BK*R)/R
      RETURN
      END
C*************************************************************
      COMPLEX FUNCTION FUNC2(E)
C*************************************************************
      COMPLEX U
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL
      COMMON /INT/ M,RHO2,ZP
      DATA PI,U /3.14159265,(0.,1.)/
      R=SQRT(RHO2+(E-ZP)**2)
      FUNC2=CEXP(-U*(M*PI*E/BL+BK*R))/R
```

```
      RETURN
      END
C*****************************************************
      SUBROUTINE PLOTR(X,Y,N,NR)
C*************************************************************
C
C  WRITTEN 2/14/74   BY J. M. PUTNAM       DEPT 220     X23877
C
C  THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C     N IS THE NUMBER OF POINTS TO BE PLOTTED.
C     NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C     NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C        NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C
      REAL X(1),Y(1),HEAD(10)
      INTEGER  LINE(101),BLANK,STAR,PLUS
      DATA BLANK,STAR,PLUS /1H ,1H*,1H+/
      NC=51
      N10=(NC-1)/10
      WRITE(6,500)
500   FORMAT(//,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
      WRITE(6,504)
      XMIN=X(1)
      XMAX=X(1)
      YMIN=Y(1)
      YMAX=Y(1)
      DO 6 I=1,N
      IF(X(I).LT.XMIN) XMIN=X(I)
      IF(X(I).GT.XMAX) XMAX=X(I)
      IF(Y(I).LT.YMIN) YMIN=Y(I)
      IF(Y(I).GT.YMAX) YMAX=Y(I)
6     CONTINUE
      DEL=XMAX-XMIN
      IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
      XMAX=XMIN+DEL
      YMAX=YMIN+DEL
      DO 5 I =1,N10
      Z=I
5     HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
      DY=(YMAX-YMIN)/(NR-1)
      IEDGE=1
      IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
      Z=YMAX+DY
      YL=Z-DY/2.
      DO 7 J=1,NR
      DO 8 K=1,NC
8     LINE(K)=BLANK
      Z=Z-DY
      YU=YL
      YL=Z-DY/2.
      DO 9 I=1,N
      IF(Y(I).GE.YU) GO TO 9
```

```
C6957         IF(Y(I).LT.YL) GO TO 9
C6967         K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
C6977         IF(K.GT.NC) K=NC
C6987         LINE(K)=STAR
C6997         IF(MCD(I,2).EQ.1) LINE(K)=PLUS
C7007         IF(IEDGE.EQ.0) GO TO 9
C7017         IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
C7027 9       CONTINUE
C7037         WRITE(6,508) Z,(LINE(K),K=1,NC)
C7047 7       CONTINUE
C7057         WRITE(6,504)
C7067         WRITE( 6,3002)
C7077         WRITE( 6,507) XMIN,(HEAD(I),I=1,N10)
C7087         RETURN
C7097  504    FORMAT ( 1X, 14(1H-), 1H., 10(5H----.), 1H- )
C7107  507    FORMAT(10X,11(F10.4))
C7117  508    FORMAT (1X, F12.4,1X, 1HI, 51A1, 1HI )
C7127 3002    FORMAT(4X,7HYH / XH,4X,1HI,5(9X,1HI))
C7137         END
C7140C ---------------------------------------------------------------
C7150         SUBROUTINE ZLIST(NI,NJ,LI,LJ,NAMI,NAMJ,Z)
C7160C WRITE Z TO THE LINE-PRINTER BY SUBMATRIX BY COLUMNS.
C7170         COMPLEX Z(1)
C7180         DIMENSION LI(1),LJ(1),NAMI(1),NAMJ(1)
C7190         NCOL=0
C7200         DO 100 I=1,NI
C7210 100     NCOL=NCOL+LI(I)
C7220         NR=1
C7230         DO 300 J=1,NJ
C7240         IF(LJ(J).EQ.0) GO TO 300
C7250         NC=1
C7260         DO 200 I=1,NI
C7270         IF(LI(I).EQ.0) GO TO 200
C7280         WRITE(6,1) NAMI(I),NAMJ(J)
C7290 1       FORMAT(/,2H Z,1X,A4,3H - ,A4)
C7300         KI=LI(I)
C7310         KJ=LJ(J)
C7320         K1=(NR-1)*NCOL+NC
C7330         DO 150 KK=1,KJ
C7340         K2=K1+KI-1
C7350         WRITE(6,2) (Z(K),K=K1,K2)
C7360 2       FORMAT(1X,10G11.4)
C7370 150     K1=K1+NCOL
C7380         NC=NC+LI(I)
C7390 200     CONTINUE
C7400         NR=NR+LJ(J)
C7410 300     CONTINUE
C7420         RETURN
C7430         END
C7440/EOR
```



```
CC100BOTZ,P30,T100,CM120000,STCX3.
C0110ACCOUNT,M0326S,BOT2A.
001208ANNERS(OUTPUT) *J. PUTNAM*DEPT 220*BLD 110-4**
C0130DEFINE(OUTFIL=ZWWX)
C0138FIN(R=C,OPT=2)
00140LDSET(PRESET=INDEF)
C015CLGO.
C021CEXIT.
00230/EOR
C0240       PROGRAM BOTZWW(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C0250      1 OUTFIL,TAPE7=OUTFIL)
00271C
C0272C   WIRE WIRE-JUNCTION INTERACTION MATRIX ELEMENTS
00273C   JUNCTION JUNCTION INTERACTION MATRIX ELEMENTS
C0274C   RELEVANCE: STRONG COUPLING WIRE-BOR ANALYSIS
00275C   J.F.SHAEFFER
C0276C   MCDONNELL DOUGLAS RESEARCH LABORATORIES
C0277C   SAINT LOUIS, MISSOURI
C0280C
CC290C   JUNCTION MODELED AS HALF TRIANGLE FUNCTION, 1ST PT IS THE MAXIMUM
00300CC
CC302        COMPLEX Z(3600)
CC305        COMMON /WAVE/ BK
C0310        COMMON /WIRE1/ NPW,X(110,3)
C0312        COMMON /WIRE2/ WL(110)
C0314        COMMON /WIRE3/ NW,INDW(6),RADW(5)
00316        COMMON /WIRE4/ TWIRE(110,4),TPW(110,4),UW(110,3)
C0318        COMMON /JUNC1/ NJ,PADJ(10),RADD(10)
00320        COMMON /JUNC2/ TJUNC(10,2),TPJ(10,2)
00322        COMMON /JUNC3/ XJ(10,3,3),WLJ(10,2)
CC324        COMMON /JUNC4/ UJ(10,2,3)
00326        COMMON /JUNC5/ UPT(10,3)
C0328        COMMON /JUNC6/ URZ(10,3)
0C35C        COMMON/ CONST / PI,ETA,IMAG
C0360        COMPLEX IMAG,PSI
C039C        COMPLEX EJKR
00400        DIMENSION SPV(3),EPV(3),RO(3)
C0430        DIMENSION U(3),XJJ(3),XIT(3)
CC440        INTEGER P,Q
C0452        DIMENSION LI(2),LJ(2),NAMI(2),NAMJ(2)
CC454        DATA NAMI,NAMJ /4HWIRE,4HJUNC,4HWIRE,4HJUNC/
00460        DATA PI,PERM,EO/3.1415927,1.2566E-6,8.85E-12/
CC470        DATA C,ETA/2.99792E8,377.0/
CC480C
00490        IMAG=(C,C,1.C)
CC492        READ(5,1) BK
C0494 1      FORMAT(E15.7)
CC496        WRITE(6,2) BK
C0498 2      FORMAT(9H1       BK,/,E15.7)
00510C
C052C        CALL DATAIN
00530C
```

```
00535       WRITE(7) NW,NPW,NJ
00540       W=C*BK
00555C  COMPUTE WIRE JUNCTION MATRIX ELEMENTS
00560       NWIRE=(NPW-NW*3)/2
00565       IF(NJ.EQ.0) GO TO 167
00570C
00580       DO 10 IJUNC=1,NJ
00590C
00600       ISTOP=0
00610       DO 10 KWIRE=1,NW
00620C
00630       ISTART=ISTOP+1
00640       ISTOP=(INDW(KWIRE+1)-INDW(KWIRE)-3)/2 + ISTOP
00650C
00660       DO 10 IWIRE=ISTART,ISTOP
00670C
00690C
00695       L1=(NWIRE+IJUNC-1)*(NWIRE+NJ)+IWIRE
00697       L1T=(IWIRE-1)*(NWIRE+NJ)+NWIRE+IJUNC
00700       Z(L1)=0.0
00710       KSW=2*(IWIRE-1) + 3*(KWIRE-1)
00720C
00730C  WIRE-SURFACE JUNCTION IMPEDANCE
00740C
00750C  SURFACE DISK RADIUS, B
00770       B=RADD(IJUNC)
00780       F1=BK*ETA/(4.*PI*2.*PI*B)
00790       INCR=2
00792       INCA=4
00800       DR=B/INCR
00802       DALPHA=2.*PI/INCA
00810C
00820       DO 11 P=1,4
00830       KS=KSW+P
00840C
00850       DO 11 IR=1,INCR
00860       R=(IR-1)*DR+DR/2.
00870C
00880       DO 11 IA=1,INCA
00890       ALPHA=(IA-1)*DALPHA + DALPHA/2.
00900C
00910       SA=SIN(ALPHA)
00912       CA=COS(ALPHA)
00920C
00930C  COMPUTE SURFACE DISK UNIT VECTOR, SOURCE POINT COORDINATES,
00940C   DOT PRODUCT R.L, SOURCE-FIELD DISTANCE
00950C
00960       DOT1=0.0
00962       RR=0.0
00970       DO 12 I=1,3
00980       RD(I)=CA*URT(IJUNC,I) + SA*URZ(IJUNC,I)
00990       DOT1=DOT1 + RD(I)*UW(KS,I)
```

```
C1000       SPV(I)=XJ(IJUNC,1,I) + P*RD(I)
C1010       XR=SPV(I) - 0.5*(X(KS+1,I)+X(KS,I))
C1020  12    RR=RR + XR*XR
C1030       RR=SQRT(RR)
C1040C
C1048       ARG=BK*RR
C1050       EJKR=(COS(ARG) - IMAG*SIN(ARG)) / RR
C1060C
C1070       FACT=F1*WL(KS)*DR*DALPHA
C1080C
C1090       Z(L1)=Z(L1) + (TWIRE(IWIRE,P)*(R-B)*DOT1
C1100      1                -TPW(IWIRE,P)/BK/BK)*IMAG*EJKR*FACT
C1120  11   CONTINUE
C1140C
C1150C
C1160C  WIRE-WIRE JUNCTION IMPEDANCE
C1170C
C1180       DO 5 P=1,4
C1190       KS=KSW+P
C1200C
C1210       DO 5 Q=1,2
C1220       DOT=0.0
C1230       DO 4 K=1,3
C1240       DOT=DOT+UW(KS,K)*UJ(IJUNC,Q,K)
C1250       XJJ(K)=0.5*(XJ(IJUNC,Q+1,K) + XJ(IJUNC,Q,K))
C1260       XII(K)=0.5*(X(KS+1,K) + X(KS,K))
C1270  4    U(K)=UW(KS,K)
C1280       DLJ=WL(KS)
C1290C
C1300       CALL GREEN(PSI,XJJ,XII,U,DLJ,BK,RADW(KWIRE))
C1310C
C1320       Z(L1)=Z(L1) + (PERM*W*DOT*TWIRE(IWIRE,P)
C1330      1*TJUNC(IJUNC,Q) - TPW(IWIRE,P)*TPJ(IJUNC,Q)/W/E0)
C1340      2*IMAG*PSI*WLJ(IJUNC,Q)
C1350  5    CONTINUE
C1365       Z(L1T)=Z(L1)
C1370  10   CONTINUE
C1380C
C1390C  COMPUTE JUNCTION-JUNCTION MATRIX ELEMENTS
C1400C
C1410       DO 100 IJUNC=1,NJ
C1420C
C1430       DO 100 JJUNC=1,NJ
C1435       I=IJUNC
C1437       J=JJUNC
C1440C
C1450       L1=(NWIRE+JJUNC-1)*(NWIRE+NJ)+NWIRE+IJUNC
C1452       Z(L1)=0.0
C1460C
C1470       DO 19 Q=1,2
C1490       DO 19 P=1,2
C1500C
```

```
01510       DOT=0.0
01520       DO 18 K=1,3
01530       DOT=DOT + UJ(I,P,K)*UJ(J,Q,K)
01540       XJJ(K)=0.5*(XJ(J,Q+1,K) + XJ(J,Q,K))
01550       XII(K)=0.5*(XJ(I,P+1,K) + XJ(I,P,K))
01560 18    U(K)=UJ(I,P,K)
01570       DLJ=WLJ(I,P)
01580C
01590       CALL GREEN(PSI,XJJ,XII,U,DLJ,BK,RADJ(I))
01600C
01610       Z(L1)=Z(L1) + (PERM*W*DOT*TJUNC(I,P)*TJUNC(J,Q)
01620      1-TPJ(I,P)*TPJ(J,Q)/W/EO) * IMAG*PSI*WLJ(J,Q)
01630 19    CONTINUE
01640 20    CONTINUE
01650C
01660C   HALF TRIANGLE-SURFACE DISK IMPEDANCE
01670C
01690C   JJUNC--HALF TRIANGLE
01700C
01720C   IJUNC--SURFACE DISK
01725C
01726       IF(IJUNC.EQ.JJUNC) GO TO 115
01730C
01731       MJUNC=IJUNC
01732       NJUN=JJUNC
01733       DO 110 M=1,2
01735       IF(M.EQ.2) MJUNC=JJUNC
01737       IF(M.EQ.2) NJUN=IJUNC
01738       A=RADJ(MJUNC)
01740C
01750C   SURFACE DISK RADIUS, B
01770       B=RADD(MJUNC)
01780       F1=BK*ETA/(4.*PI*2.*PI*(B-A))
01800       INCR=2
01802       INCA=4
01810       DR=(B-A)/INCR
01812       DALPHA=2.*PI/INCA
01820C
01830       DO 110 Q=1,2
01850       DO 110 IR=1,INCR
01860       R=A + (IR-0.5)*DR
01870C
01880       DO 110 IA=1,INCA
01890       ALPHA=(IA-1)*DALPHA + DALPHA/2.
01900       SA=SIN(ALPHA)
01902       CA=COS(ALPHA)
01920C
01930C   COMPUTE SURFACE DISK UNIT VECTOR, SOURCE POINT COORDINATES,
01940C    DOT PRODUCT R.L, SOURCE-FIELD DISTANCE
01950C
01960       DOT1=0.0
01962       RR=0.0
```

```
01970       DO 120 I=1,3
01980       RD(I)=CA*URT(MJUNC,I) + SA*URZ(MJUNC,I)
01990       DOTI=DOTI + RD(I)*UJ(NJUN,Q,I)
02000       SPV(I)=XJ(MJUNC,1,I) + R*RD(I)
C2010       XR=SPV(I) - 0.5*(XJ(NJUN,Q+1,I)+XJ(NJUN,Q,I))
C2020 120   RR=RR + XR*XR
02030       RR=SQRT(RR)
C2040C
C2048       ARG=BK*RR
C2050       EJKR=(COS(ARG) - IMAG*SIN(ARG)) / RR
C2060C
C2070       FACT=F1*WLJ(NJUN,Q)*DR*DALPHA
C2080C
C2090       Z(LI)=Z(LI) + (TJUNC(NJUN,Q)*(R-B)*DOTI
02100      1                    -TPJ(NJUN,Q)/BK/BK)*IMAG*EJKR*FACT
02102 110   CONTINUE
02104       GO TO 116
02106 115   CONTINUE
C2108C
02110C   SELF TERM FOR WIRE-DISK (MULT BY 2)
C2112C
02114       B=RADD(IJUNC)
02115       A=RADJ(IJUNC)
02116       HO=WLJ(IJUNC,1)+WLJ(IJUNC,2)
02117       T3=0.0
02118       INCR=40
C2119       DR=(B-A)/INCR
C2120       DO 521 IR=1,INCR
02121       R=A+(IR-0.5)*DR
C2122       T3=T3+DR*ALOG((HO+SQRT(HO*HO+R*R))/R)
C2124 521   CONTINUE
02126       T3=T3/(B-A)/HO
C2128       Z(LI)=Z(LI) + 2.*(ETA/4./PI +
02133      1 IMAG*ETA/4./PI/BK*T3)
C2134C
02135 116   CONTINUE
02140C   DISK-DISK INTERACTION
C2150C
C2160C SURFACE DISK RADIUS
C2170       B1=RADD(IJUNC)
C2180       B2=RADD(JJUNC)
C2190       IF(IJUNC.NE.JJUNC) GO TO 135
C2200C
C2210C   SELF TERM FOR DISK-DISK
C2220C
C2225       INCB=40
C2227       INCR=16
C2230       B=B1
C2232       A=RADJ(IJUNC)
C2235C
C2240       DR=(B-A)/INCR
C2242       DBETA=PI/INCB
```

```
C2245C
C2248       A2=A*A
C2249       B2=B*B
C2250       FI=BK*ETA/(2.*PI*(B-A))**2
C2255       F=0.0
C2260C
C2265       DO 222 IB=1,INCR
C2270       BETA=(IB-0.5)*DBETA
C2272       CB=COS(BETA)
C2275C
C2280       DO 222 IR=1,INCR
C2285       R=A + (IR-0.5)*DR
C2286       R2=R*R
C2287       RCB=R*CB
C2290C
C2295       T1=SQRT(B2 - 2.*R*RCB + R2)
C2300       T2=SQRT(A2 - 2.*A*RCB + R2)
C2305       FI1=ALOG(( T1 + B - RCB ) / ( T2 + A - RCB))
C2310       FI2=T1 - T2 + RCB*FI1
C2315C
C2320       F=F + ((B*(B-R)*CB - 1./BK/BK)*FI1 - (B-R)*CB*FI2)*DR*DBETA
C2325C
C2330 222   CONTINUE
C2335C
C2340       Z(L1)=Z(L1) + FI*IMAG*(PI*IMAG*(R-A)**2/BK
C2341      1    + F )
C2470       GO TO 131
C2500 135   CONTINUE
C2510       FI=BK*ETA/(4.*PI)/(2.*PI)**2/B1/B2
C2520       INCR=2
C2522       INCA=4
C2530       DR1=B1/INCR
C2532       DR2=B2/INCR
C2540       DALPHA=2.*PI/INCA
C2550C
C2560       DO 130 IR1=1,INCR
C2570       R1=(IR1-0.5)*DR1
C2580C
C2590       DO 130 IR2=1,INCR
C2600       R2=(IR2-0.5)*DR2
C2610C
C2620       DO 130 IA1=1,INCA
C2630       A1=(IA1-0.5)*DALPHA
C2640       SA1=SIN(A1)
C2642       CA1=COS(A1)
C2650C
C2660       DO 130 IA2=1,INCA
C2670       A2=(IA2-0.5)*DALPHA
C2680       SA2=SIN(A2)
C2682       CA2=COS(A2)
C2690C
C2700C  COMPUTE SURFACE DISK UNIT VECTORS; SOURCE POINT-FIELD POINT
```

```
C271CC    COORDINATES & DISTANCE; DOT PRODUCT R1*R2
C2720C
C2730       DOT1=C.C
C2732       RR=0.0
C2740       DO 14C I=1,3
C2750       RD1=CA1*URT(IJUNC,I) + SA1*URZ(IJUNC,I)
C2760       RD2=CA2*URT(JJUNC,I) + SA2*URZ(JJUNC,I)
C2770       DOT1=DOT1 + RD1*RD2
C2780       SPV(I)=XJ(IJUNC,1,I) + P1*RD1
C2790       FPV(I)=XJ(JJUNC,1,I) + P2*RD2
C2800       XR=SPV(I) - FPV(I)
C2810 14C   RR=RR + XR*XR
C2815       RR=SORT(RR)
C2820C
C2828       ARG=BK*RR
C283C       EJKR=(COS(ARG) - IMAG*SIN(ARG)) / RR
C2840C
C2850       FACT=F1*DALPHA*DALPHA*DR1*DR2
C2860C
C2870       Z(L1)=Z(L1) + ((B1-R1)*(R2-R2)*DOT1
C2880      1                    -1./BK/BK)*IMAG*EJKR*FACT
C2900 130   CONTINUE
C2910 131   CONTINUE
C2920C
C2930 100   CONTINUE
C2935 167   CONTINUE
C2940C
C2950C   COMPUTE WIRE-WIRE MATRIX ELEMENTS, ZWW
C2960C
C2970       ISTOP=0
C2980C
C2990       DO 30 KWIRE=1,NW
C3000C
C3010       ISTART=ISTOP + 1
C3020       ISTOP=(INDW(KWIRE+1)-INDW(KWIRE)-3)/2 + ISTOP
C3030C
C3040       DO 3C I=ISTART,ISTOP
C3050C
C3060       JSTOP=0
C3070C
C3080       DO 3C KJWIRE=1,NW
C3090C
C3100       JSTART=JSTOP+1
C3110       JSTOP=(INDW(KJWIRE+1)-INDW(KJWIRE)-3)/2 + JSTOP
C3120C
C3130       DO 3C J=JSTART,JSTOP
C3140C
C3150       L1=(J-1)*(NWIRE+NJ)+I
C3152       Z(L1)=0.0
C3160       IK=2*(I-1) + 3*(KWIRE-1)
C3170       JK=2*(J-1) + 3*(KJWIRE-1)
C3180C
```

```
      DO 29 P=1,4
      IP=IK+P
C
      DO 29 Q=1,4
      JQ=JK+Q
      DOT=0.0
C
      DO 28 K=1,3
      DOT=DOT+UW(IP,K)*UW(JQ,K)
      XJJ(K)=0.5*(X(IP+1,K)+X(IP,K))
      XII(K)=0.5*(X(JQ+1,K)+X(JQ,K))
28    U(K)=UW(IP,K)
      DLJ=WL(IP)
C
      CALL GREEN(PSI,XJJ,XII,U,DLJ,BK,RADW(KWIRE))
C
      Z(LI)=Z(LI) + (PERM*W*DOT*TWIRE(I,P)*TWIRE(J,Q) -
     1          TPW(I,P)*TPW(J,Q)/W/EO) *
     2          IMAG*PSI*WL(JQ)
29    CONTINUE
30    CONTINUE
C PRINT OUT IMPEDANCE MATRIX Z BY SUBMATRIX BY COLUMNS.
      LI(1)=NWIRE
      LI(2)=NJ
      LJ(1)=NWIRE
      LJ(2)=NJ
      CALL ZLIST(2,2,LI,LJ,NAMI,NAMJ,Z)
C WRITE Z TO DISK.
      LWW=(NWIRE+NJ)**2
      WRITE(7) (Z(I),I=1,LWW)
      STOP
      END
C*************************************************************************
      SUBROUTINE GREEN(PSI,XJ,XI,U,DLJ,BK,RADW)
C
      COMMON/ CONST / PI,ETA,IMAG
      COMPLEX PSI,IMAG
      DIMENSION XJ(1),XI(1),U(1)
C
C  COMPUTE DISTANCE BETWEEN CENTER OF SEGMENTS XJ & XI
      WR=RADW*RADW
      RO2=0.0
      ZJ=0.0
      DO 10 K=1,3
      SD=XJ(K)-XI(K)
      RO2=RO2+SD*SD
10    ZJ=ZJ+SD*U(K)
C
C
      ZJ2=ZJ*ZJ
      RHO2=RO2 - ZJ2
      ZJ=SQRT(ZJ2)
```

```
C3890C
C3900       IF(RHO2.LT.WR) RHO2=WR
C3920       RO=SQRT(RO2)
C3922       RHO=SQRT(RHO2)
C3930       ARG=BK*RO
C3940       ALPHA=DLJ/2.
C3950C
C3960       IF(RO.GE.(10.*ALPHA)) GO TO 20
C3970C
C3980       ZP=ZJ+ALPHA
C3982       ZM=ZJ-ALPHA
C3990       ZRP=SQRT(RHO2+ZP*ZP)
C3992       ZRM=SQRT(RHO2+ZM*ZM)
C4000       IF(ZM.LE.0.0) GO TO 12
C4010       TERM=(ZP+ZRP) / (ZM+ZRM)
C4020       AI1=ALOG(TERM)
C4030       GO TO 13
C4040  12   AI1=ALOG((ZP+ZRP)*(-ZM+ZRM)/RHO2)
C4050  13   AI2=2.*ALPHA
C4060       AI3=(ZP*ZRP - ZM*ZRM + RHO2*AI1) / 2.0
C4070       AI4=AI2*RHO2 + (AI2*ALPHA*ALPHA + 3.*AI2*ZJ2) / 3.0
C4080       BK2=BK*BK
C4082       BK3=BK2*BK
C4090       PSI=(COS(ARG) - IMAG*SIN(ARG)) / (4.*PI)
C4100       PSI=PSI*(AI1 - IMAG*BK*(AI2-RO*AI1) - BK2*(AI3-2.*RO*AI2
C4110      1     +RO2*AI1)/2. + IMAG*BK3*(AI4-3.*RO*AI3+3.*RO2*AI2
C4120      2     -RO*RO2*AI1)/6.       )
C4130       RETURN
C4140C
C4150  20   CONTINUE
C4160C
C4170       ZR=ZJ/RO
C4172       ZR2=ZR*ZR
C4174       ZR4=ZR2*ZR2
C4180       AR=ALPHA/RO
C4182       AR2=AR*AR
C4184       AR3=AR2*AR
C4186       AR4=AR3*AR
C4190       AK=BK*ALPHA
C4192       AK2=AK*AK
C4194       AK3=AK2*AK
C4196       AK4=AK3*AK
C4200       A0=1.+AR2/6.*(3.*ZR2-1.0) + AR4/40.*(3.-30.*ZR2+35.*ZR4)
C4210       A1=AR/6.*(3.*ZR2-1.0) + AR3/40.*(3.-30.*ZR2+35.*ZR4)
C4220       A2=-ZR2/6. - AR2/40.*(1.-12.*ZR2+15.*ZR4)
C4230       A3=AR/60.*(3.*ZR2-5.*ZR4)
C4240       A4=ZR4/120.
C4250       PSI=(COS(ARG) - IMAG*SIN(ARG)) / (4.*PI*RO) * (2.*ALPHA)
C4260       PSI=PSI*(A0 + IMAG*AK*A1 + AK2*A2 + IMAG*AK3*A3 + AK4*A4)
C4270       RETURN
C4280C
C4290       END
```

```
C4540C ------------------------------------------------------------
C4550       SUBROUTINE DATAIN
C4560C
C4570       COMMON /WIRE1/ NPW,X(110,3)
C4572       COMMON /WIRE2/ WL(110)
C4574       COMMON /WIRE3/ NW,INDW(6),RADW(5)
C4576       COMMON /WIRE4/ TWIRE(110,4),TPW(110,4),UW(110,3)
C4578       COMMON /JUNC1/ NJ,RADJ(10),RADD(10)
C4580       COMMON /JUNC2/ TJUNC(10,2),TPJ(10,2)
C4582       COMMON /JUNC3/ XJ(10,3,3),WLJ(10,2)
C4584       COMMON /JUNC4/ UJ(10,2,3)
C4586       COMMON /JUNC5/ URT(10,3)
C4588       COMMON /JUNC6/ URZ(10,3)
C4635       DIMENSION INDJW(10),INDTJ(10),UXJ(10),UYJ(10),UZJ(10)
C4640C
C4650       DIMENSION DL(3)
C4670C
C4680       CALL SBOTIN
C4737C
C4739       CALL SCAPIN
C4740       READ(5,16) NW,NPW,NJ
C4750 16    FORMAT(3I3)
C4760C
C4780C      NPW=NO. WIRE POINTS
C4790C      NW= NO. WIRES
C4800C
C4810       WRITE(6,1) NW,NPW,NJ
C4820 1     FORMAT(26H       NW      NPW      NJ    ,/,3I8)
C4850C
C4860C   READ IN WIRE COORDINATES
C4870       DO 17 I=1,3
C4880 17    READ(5,12) (X(J,I),J=1,NPW)
C4885 12    FORMAT(10F8.4)
C4890C
C4900C   INDW(I)=START POINT FOR EACH WIRE
C4910       READ(5,18) (INDW(I),I=1,NW)
C4915 18    FORMAT(10I8)
C4920       INDW(NW+1)=NPW + 1
C4930C
C4932       READ(5,12) (RADW(I),I=1,NW)
C4934C   READ JUNCTION COORDINATES
C4936       IF(NJ.EQ.0) GO TO 50
C4938       READ(5,18) (INDJW(I),I=1,NJ)
C4940       READ(5,12) (RADD(I),I=1,NJ)
C4942       READ(5,12) (UXJ(I),I=1,NJ)
C4944       READ(5,12) (UYJ(I),I=1,NJ)
C4946       READ(5,12) (UZJ(I),I=1,NJ)
C4948 50    IERW=0
C4950       IERJ=0
C4952       WRITE(6,61)
C4954 61    FORMAT(/,22X,'WIRE COORDINATES',20X,'JUNCTION PARAMETERS',/,
C4956      1 26X,'IW',4X,'XW',6X,'YW',6X,'ZW',
```

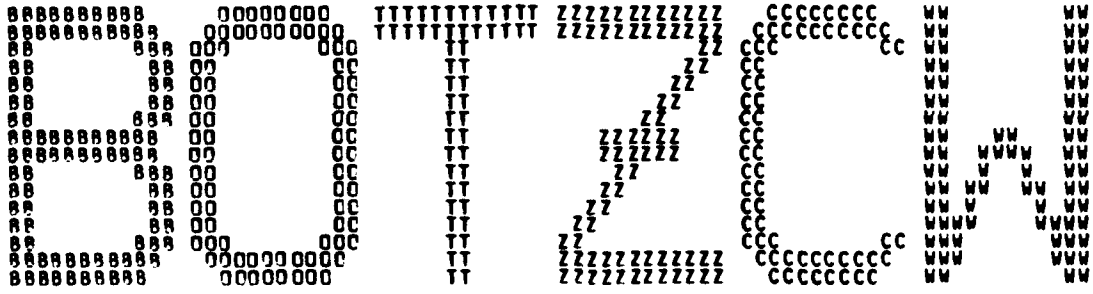
```
C4958      2 10X,'IJ',4X,'RADJ',5X,'UXJ',5X,'UYJ',5X,'UZJ')
C4960        IJ=1
C4962C THIS LOOP LISTS WIRE/JUNCTION POINTS, WHILE CHECKING THE FOLLOWING:
C4964C    1) EACH WIRE MUST CONTAIN AN ODD NUMBER OF POINTS.
C4966C    2) EACH JUNCTION MUST EITHER START OR TERMINATE A WIRE.
C4968C    3) CHECK THAT ALL JUNCTION POINTS ARE FOUND.
C4970        DO 100 IW=1,NW
C4972        WRITE(6,62) IW,RADW(IW)
C4974 62     FORMAT(2X,'WIRE',I3,' RADW=',F8.4)
C4976        I1=INDW(IW)
C4978        I2=INDW(IW+1)-1
C4980C CHECK FOR AN ODD NUMBER OF POINTS ON WIRE IW.
C4982        IF(MOD(I2-I1+1,2).NE.1) IERW=1
C4984        DO 90 I=I1,I2
C4986        WRITE(6,63) I,X(I,1),X(I,2),X(I,3)
C4988 63     FORMAT(25X,I3,3F8.4)
C4990C CHECK IF WIRE POINT I IS A JUNCTION POINT.
C4992        IF(IJ.GT.NJ) GO TO 90
C4994        IF(INDJW(IJ).NE.I) GO TO 90
C4996C CHECK THAT JUNCTION POINT IJ IS AT THE START OR END OF WIRE.
C4998        IF(I.NE.I1 .AND. I.NE.I2) IERJ=1
C5000        WRITE(6,64) IJ,RADJ(IJ),UXJ(IJ),UYJ(IJ),UZJ(IJ)
C5002 64     FORMAT('+',58X,I3,4F8.4)
C5004C DETERMINE DIRECTION IN WHICH WIRE LEAVES JUNCTION POINT.
C5006        INDTJ(IJ)=1
C5008        IF(I.EQ.I2) INDTJ(IJ)=-1
C5010C COMPUTE JUNCTION PARAMETERS.
C5012        RADJ(IJ)=RADW(IW)
C5014        DO 105 J=1,3
C5015        XJ(IJ,1,J)=X(I,J)
C5016        XJ(IJ,2,J)=X(I+INDTJ(IJ),J)
C5017        XJ(IJ,3,J)=X(I+2*INDTJ(IJ),J)
C5018 105    CONTINUE
C5020        IJ=IJ+1
C5022 90     CONTINUE
C5024 100    CONTINUE
C5026C CHECK FOR WIRE OR JUNCTION INPUT ERRORS.
C5028        IF(IERW.NE.0) GO TO 980
C5030        IF(IJ-1.NE.NJ .OR. IERJ.NE.0) GO TO 990
C5380C
C5390C   WIRE PARAMETERS: UNIT VECTOR, UW(I) ; LENGTH, WL(NSEG)
C5400        NSTOP=0
C5410C
C5440        DO 40 KWIRE=1,NW
C5450        ISTART=INDW(KWIRE)
C5452        ISTOP=INDW(KWIRE+1)-2
C5460C
C5470        DO 35 NSEG=ISTART,ISTOP
C5480        IP=NSEG+1
C5490        DO 36 I=1,3
C5500 36     DL(I)= X(IP,I)-X(NSEG,I)
C5510C
```

```
C5520       WL(NSEG)=SQRT( DL(1)*DL(1)+DL(2)*DL(2)+DL(3)*DL(3) )
C5530C
C5540       DO 34 I=1,3
C5550 34    UW(NSEG,I)=DL(I)/WL(NSEG)
C5560C
C5580C
C5590 35    CONTINUE
C5600C   COMPUTE TRIANGLE FUNCTIONS FOR EACH WIRE
C5610       NSTART=NSTOP+1
C5620       NSTOP=NSTOP + (ISTOP-ISTART-1)/2
C5630       DO 38 N=NSTART,NSTOP
C5640       J1=2*(N-1)+1+3*(KWIRE-1)
C5650       J2=J1+1
C5652       J3=J2+1
C5654       J4=J3+1
C5660       DEL1=WL(J1)+WL(J2)
C5662       DEL2=WL(J3)+WL(J4)
C5670       TWIRE(N,1)=0.5*WL(J1)/DEL1
C5680       TWIRE(N,2)=(WL(J1)+0.5*WL(J2))/DEL1
C5690       TWIRE(N,3)=(WL(J4)+0.5*WL(J3))/DEL2
C5700       TWIRE(N,4)=0.5*WL(J4)/DEL2
C5710       TPW(N,1)=1./DEL1
C5712       TPW(N,2)=1./DEL1
C5720       TPW(N,3)=-1./DEL2
C5722       TPW(N,4)=-1./DEL2
C5730 38    CONTINUE
C5740 40    CONTINUE
C5750C
C5940       IF(NJ.EQ.0) RETURN
C5950C   COMPUTE JUNCTION UNIT VECTORS AND SEGMENT LENGTHS
C5960       DO 116 N=1,NJ
C5970       DO 116 J=1,2
C5980       DO 114 I=1,3
C5990 114   DL(I)=XJ(N,J+1,I)-XJ(N,J,I)
C6000       WLJ(N,J)=SQRT(DL(1)*DL(1)+DL(2)*DL(2)+DL(3)*DL(3))
C6010       DO 115 I=1,3
C6020 115   UJ(N,J,I)=DL(I)/WLJ(N,J)
C6030 116   CONTINUE
C6040C
C6050C   COMPUTE HALF TRIANGLE FUNCTION CENTROID VALUES
C6060       DO 120 N=1,NJ
C6070       DEL2=WLJ(N,1)+WLJ(N,2)
C6080       TPJ(N,1)=-1./DEL2
C6082       TPJ(N,2)=-1./DEL2
C6090       TJUNC(N,1)=(WLJ(N,2)+0.5*WLJ(N,1))/DEL2
C6100 120   TJUNC(N,2)=0.5*WLJ(N,2)/DEL2
C6110C COMPUTE UNIT VECTORS FOR JUNCTION(DISK).
C6112       DO 95 IJ=1,NJ
C6114C NORMAL UNIT VECTOR (UJ).
C6116       RR=SQRT(UXJ(IJ)**2+UYJ(IJ)**2+UZJ(IJ)**2)
C6118       UXJ(IJ)=UXJ(IJ)/RR
C6120       UYJ(IJ)=UYJ(IJ)/RR
```

```
C6122       UZJ(IJ)=UZJ(IJ)/RR
C6124C FIND 2 ORTHOGONAL UNIT VECTORS IN THE PLANE OF THE DISK (URT & URZ).
C6126       URT(IJ,1)=0.C
C6128       URT(IJ,2)=0.0
C613C       URT(IJ,3)=0.0
C6132       URZ(IJ,1)=C.C
C6134       URZ(IJ,2)=0.0
C6136       URZ(IJ,3)=0.0
C6138       IF(UXJ(IJ).EQ.0.0) GO TO 91
C6140C FIND INTERSECTION WITH X-Y PLANE.
C6142       URT(IJ,2)=1.C
C6144       URT(IJ,1)=-UYJ(IJ)/UXJ(IJ)
C6146C FIND INTERSECTION WITH X-Z PLANE.
C6148       URZ(IJ,3)=1.C
C6150       URZ(IJ,1)=-UZJ(IJ)/UXJ(IJ)
C6152       GO TO 94
C6154 91    IF(UYJ(IJ).EQ.0.0) GO TO 92
C6156C FIND INTERSECTION WITH Y-Z PLANE.
C6158       URT(IJ,3)=1.0
C6160       URT(IJ,2)=-UZJ(IJ)/UYJ(IJ)
C6162C FIND INTERSECTION WITH X-Y PLANE.
C6164       URZ(IJ,1)=1.C
C6166       URZ(IJ,2)=-UXJ(IJ)/UYJ(IJ)
C6168       GO TO 94
C6170 92    IF(UZJ(IJ).EQ.0.0) GO TO 94
C6172C FIND INTERSECTION WITH X-Z PLANE.
C6174       URT(IJ,1)=1.0
C6176       URT(IJ,3)=-UXJ(IJ)/UZJ(IJ)
C6178C FIND INTERSECTION WITH Y-Z PLANE.
C6180       URZ(IJ,2)=1.0
C6182       URZ(IJ,3)=-UYJ(IJ)/UZJ(IJ)
C6184 94    CONTINUE
C6186       RR=SQRT(URT(IJ,1)**2+URT(IJ,2)**2+URT(IJ,3)**2)
C6188       URT(IJ,1)=URT(IJ,1)/RR
C6190       URT(IJ,2)=URT(IJ,2)/RR
C6192       URT(IJ,3)=URT(IJ,3)/RR
C6194       RR=SQRT(URZ(IJ,1)**2+URZ(IJ,2)**2+URZ(IJ,3)**2)
C6196       URZ(IJ,1)=URZ(IJ,1)/RR
C6198       URZ(IJ,2)=URZ(IJ,2)/RR
C6200       URZ(IJ,3)=URZ(IJ,3)/RR
C6206 95    CONTINUE
C6290       RETURN
C6300 980   WRITE(6,981)
C6310 981   FORMAT(//,' **** ERROR IN WIRE INPUT')
C6320       STOP
C6330 990   WRITE(6,991)
C6340 991   FORMAT(//,' **** ERROR IN JUNCTION INPUT')
C6350       STOP
C6390       END
C6580C ------------------------------------------------------------
C6590       SUBROUTINE SBOTIN
C6600C
```

```
C6610C SKIP BOT INPUTS.
C6620C       READ(5,49) NMODE,NPT,NBAND
C6630C       READ(5,49) NP
C6640 49     FORMAT(3I3)
C6650C       READ(5,53)(YB,I=1,NP)
C6760C       READ(5,53)(XB,I=1,NP)
C6770    53  FORMAT(10F8.4)
C6870C       READ(5,53) BL
C7600C       RETURN
C7610C       END
C7620C ------------------------------------------------
C7630        SUBROUTINE SCAPIN
C7640C
C7650C SKIP CAP INPUTS.
C7660C
C7670        READ(5,1) NC,NPR,NE
C7680 1      FORMAT(3I3)
C7690C       IF(NC.EQ.0) RETURN
C7700        IF(NE.NE.0) NE=NC
C7710C       READ(5,2) XC,YC
C7720 2      FORMAT(10F8.4)
C7730        READ(5,2) (ZC,I=1,NC)
C7740C       READ(5,2) (RHOC,I=1,NPR)
C7750        IF(NE.EQ.0) RETURN
C7760        READ(5,2) (ZE,I=1,NE)
C7770C       RETURN
C7780        END
C7785C ------------------------------------------------------
C7790        SUBROUTINE ZLIST(NI,NJ,LI,LJ,NAMI,NAMJ,Z)
C7800C WRITE Z TO THE LINE-PRINTER BY SUBMATRIX BY COLUMNS.
C7810        COMPLEX Z(1)
C7820        DIMENSION LI(1),LJ(1),NAMI(1),NAMJ(1)
C7830        NCOL=0
C7840C       DO 1CC I=1,NI
C7850C 1CC   NCOL=NCOL+LI(I)
C7860C       NR=1
C7870C       DO 3CC J=1,NJ
C7880C       IF(LJ(J).EQ.0) GO TO 300
C7890C       NC=1
C7900C       DO 2CC I=1,NI
C7910C       IF(LI(I).EQ.0) GO TO 200
C7920C       WRITE(6,1) NAMI(I),NAMJ(J)
C7930 1      FORMAT(/,2H Z,1X,A4,3H - ,A4)
C7940C       KI=LI(I)
C7950C       KJ=LJ(J)
C7960C       K1=(NR-1)*NCOL+NC
C7970C       DO 150 KK=1,KJ
C7980C       K2=K1+KI-1
C7990C       WRITE(6,2) (Z(K),K=K1,K2)
C8000 2      FORMAT(1X,1CG11.4)
C8010C 150   K1=K1+NCOL
C8020C       NC=NC+LI(I)
```

```
C8030  200  CONTINUE
08040       NR=NR+LJ(J)
C8050  300  CONTINUE
C8060       RETURN
08070       END
08080/EOR
```


BOTTOM

```
C01CCBOTZ,P3C,T3C,CM120000,STCX3.
C011CACCOUNT,M0326S,BOT2A.
C0120BANNERS(OUTPUT) *J.  PUTNAM*DEPT 220*RLD 110-4**
C0130DEFINE(OUTFIL=ZCWX)
C0138FIN(R=6,OPT=2)
C0140LDSET(PRESET=INDEF)
C0150LGO.
C0210EXIT.
C0230/EOR
C0240       PROGRAM BOTZCW(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C0250      1 OUTFIL,TAPE7=OUTFIL)
C0252C
C0254C BOTZCW GENERATES THE CAP-WIRE/JUNCTION MATRIX.
C0256C
C0258C UNIT 5 IS THE CARD READER.
C0260C UNIT 6 IS THE LINE PRINTER.
C0262C UNIT 7 IS A DISK FILE FOR OUTPUT OF THE Z MATRIX.
C0264C
C0280       COMPLEX U,A3,Z(6400)
C0282       COMMON /WAVE/ BK
C0284       COMMON /BOT1/ NMODE,NPT,NBAND
C0286       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C0288       COMMON /BOT3/ DH(82),SV(82),CV(82)
C0290       COMMON /BOT5/ T(160),TP(160),TZ(160)
C0291       COMMON /BOT6/ IEDGE,IUNIF
C0292       COMMON /CAP1/ NC,XC,YC,ZC(2)
C0294       COMMON /CAP2/ NPR,PHOC(21),RHOC1(20),DRHOC(20)
C0296       COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
C0298       COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
C0300       COMMON /EDG1/ NE,ZE(2),ZBE(10)
C0302       COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
C0304       COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
C0306      1 XW1(100),YW1(100),ZW1(100)
C0308       COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
C0310       COMMON /WIRE3/ NW,INDW(6),RADW(5)
C0312       COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
C0314       COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
C0316       COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
C0317       COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
C0319       COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
C0320       COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
C0330       DIMENSION LI(3),LJ(2),NAMI(3),NAMJ(2)
C0340       DATA NAMI,NAMJ /4HC(T),4HC(R),4HEDGE,4HWIRE,4HJUNC/
00430       U=(0.,1.)
C0432       PI=3.14159265
C0434       ETA=376.707
00435       READ(5,1) BK
00436  1    FORMAT(E15.7)
C0437       WRITE(6,2) BK
00438  2    FORMAT(9H1       BK,/,E15.7)
C0440C
00450       CALL BOTIN
```

```
00460C
00800       CALL CAPIN
00835C
00836       CALL WIREIN
00837C
C0838       WRITE(7) NC,NPR,NE,NW,NPW,NJ
00840       NM=(NP-3)/2
00842       NM2=NM*2
00844       LR=(NPR-3)/2
00846       LC=NC*NM*LR
00848       LC2=LC*2
00850       LE=NE*NM
00854       BKL=BK*BL
00856       BK2=BK*BK
00858       LCWJ=(LC2+LE)*(LW+NJ)
00860C CONSTANTS USED IN SUMMATION OF IMPEDANCE ELEMENTS.
00865       C1=BK*ETA
00870       C2=C1
00890C *** END INITIALIZATION OF PARAMETERS ***
02005C COMPUTATION OF IMPEDANCE ELEMENTS.
02007C CAP-WIRE LOOP.
02010       DO 30 J=1,LW
02020       L1=(J-1)*(LC2+LE)
02030       J3=(J-1)*4
02040       J1=INDTW(J)-1
02050       DO 31 IC=1,NC
02051       DO 31 IP=1,NM
02052       DO 31 IR=1,LR
02055C COMPUTE SUBMATRIX INDICES.
02060       L1=L1+1
02070       L2=L1+LC
02100       Z(L1)=0.
02110       Z(L2)=0.
02170       DO 70 JJ=1,4
02180       J2=J1+JJ
02190       J4=J3+JJ
02192       IP2=2*(IP-1)
02194       IP7=4*(IP-1)
02196       DO 71 IIP=1,4
02198       IP2=IP2+1
02200       IP7=IP7+1
02202       IR2=2*(IR-1)
02204       IR7=4*(IR-1)
02206       DO 71 IIR=1,4
02208       IR2=IR2+1
02210       IR7=IR7+1
02212       AI=AC(IP2)*ABS(RHOC(IR2+1)**2-RHOC(IR2)**2)
02214       RI=RHOC1(IR2)*RC1(IP2)
02216       XX=XW1(J2)-(XC+RHOC1(IR2)*RC1(IP2)*CPC(IP2))
02218       YY=YW1(J2)-(YC+RHOC1(IR2)*RC1(IP2)*SPC(IP2))
02220       ZZ=ZW1(J2)-ZC(IC)
02222       RHO2=XX*XX+YY*YY+ZZ*ZZ
```

```
C2224        RR=SQRT(RHO2)
C2226        A3=CEXP(-U*BK*RR)/4./PI/RR
C2295C CAP-WIRE IMPEDANCE ELEMENTS COMPUTED.
C2297C Z(L1) = ZT , Z(L2) = ZR.
C2300          Z(L1)=Z(L1)+AT*((CV(IP2)*DXW(J2)+SV(IP2)*DYW(J2))*
C2305       1  T(IP7)*TW(J7)-DHW(J2)*TP(IP7)*TPW(J7)/RHOC1(IR2)
C2310       2  /BK2)*TCT(IR7)*A3
C2315          Z(L2)=Z(L2)+AT*((CPC(IP2)*DXW(J2)+SPC(IP2)*DYW(J2))*
C2320       1  TCR(IR7)*TW(J7)-DHW(J2)*(TPCR(IR7)/RC1(IP2)+
C2325       2  TCR(IR7)/RI)*TPW(J7)/BK2)*TZ(IP7)*A3
C2370     71 CONTINUE
C2375     70 CONTINUE
C2381        Z(L1)=U*C1*Z(L1)
C2382        Z(L2)=U*C2*Z(L2)
C2406 31     CONTINUE
C2407 30     CONTINUE
C2410        IF(NJ.EQ.0) GO TO 200
C2415C NOTE, CAP-JUNCTION NOT INCLUDED.
C2417C CAP-JUNCTION LOOP.
C2427        DO 130 J=1,NJ
C2437        L1=(LW+J-1)*(LC2+LE)
C2460        DO 131 IC=1,NC
C2462        DO 131 IP=1,NM
C2464        DO 131 IR=1,LR
C2477C COMPUTE SUBMATRIX INDICES.
C2487        L1=L1+1
C2497        L2=L1+LC
C2507        Z(L1)=0.
C2517        Z(L2)=0.
C2627C CAP-JUNCTION(WIRE) IMPEDANCE ELEMENTS COMPUTED.
C2629C Z(L1) = ZT , Z(L2) = ZR.
C2712C CAP-JUNCTION(DISK) IMPEDANCE ELEMENTS COMPUTED.
C2713C Z(L1) = ZT , Z(L2) = ZR.
C2724 131    CONTINUE
C2726 130    CONTINUE
C2728 200    CONTINUE
C2729        IF(NE.EQ.0) GO TO 900
C2730C EDGE-WIRE LOOP.
C2740        DO 330 J=1,LW
C2750        L2=(J-1)*(LC2+LE)+LC2
C2760        J3=(J-1)*4
C2770        J1=INDTW(J)-1
C2780        DO 331 IC=1,NC
C2790        DO 331 IP=1,NM
C2810C COMPUTE SUBMATRIX INDICES.
C2830        L2=L2+1
C2850        Z(L2)=0.
C2860        DO 370 JJ=1,4
C2870        J2=J1+JJ
C2880        J7=J3+JJ
C2890        IP2=2*(IP-1)
C2900        IP7=4*(IP-1)
```

```
C2910       DO 371 IIP=1,4
C2920       IP2=IP2+1
C2930       IP7=IP7+1
C2940       IR2=2*LR
C2950       IR7=2*(IC-1)
C2952       IZ2=2*(IC-1)
C2954       IZ7=2*(IC-1)
C2960       DO 371 IIRZ=1,2
C2970       IR2=IR2+1
C2980       IR7=IR7+1
C2982       IZ2=IZ2+1
C2984       IZ7=IZ7+1
C2990       AI=AC(IP2)*ABS(RHOC(IR2+1)**2-RHOC(IR2)**2)
C3000       RI=RHOC1(IR2)*RC1(IP2)
C3010       XX=XW1(J2)-(XC+RHOC1(IR2)*RC1(IP2)*CPC(IP2))
C3020       YY=YW1(J2)-(YC+RHOC1(IR2)*RC1(IP2)*SPC(IP2))
C3030       ZZ=ZW1(J2)-ZC(IC)
C3040       RHO2=XX*XX+YY*YY+Z7*ZZ
C3050       RR=SQRT(RHO2)
C3060       A3=CEXP(-U*BK*RR)/4./PI/RR
C3070C EDGE(CAP)-WIRE IMPEDANCE ELEMENTS COMPUTED.
C3120       Z(L2)=Z(L2)+AI*((CPC(IP2)*DXW(J2)+SPC(IP2)*DYW(J2))*
C3130      1 TCE(IR7)*TW(J7)-DHW(J2)*(TPCE(IR7)/RCI(IP2)+
C3131      2 TCE(IR7)/RI)*TPW(J7)/BK2)*TZ(IP7)*A3
C3132       AI=DH(IP2)*ABS(ZE(IC)-ZC(IC))/2.
C3133       XX=XW1(J2)-XBI(IP2)
C3134       YY=YW1(J2)-YBI(IP2)
C3135       ZZ=ZW1(J2)-ZBE(IZ2)
C3136       RHO2=XX*XX+YY*YY+ZZ*ZZ
C3137       RR=SQRT(RHO2)
C3138       A3=CEXP(-U*BK*RR)/4./PI/RR
C3142C EDGE(BOT)-WIRE IMPEDANCE ELEMENTS COMPUTED.
C3144       Z(L2)=Z(L2)+AI*(DZW(J2)*TBE(IZ7)*TW(J7)-
C3146      1 DHW(J2)*TPBE(IZ7)*TPW(J7)/BK2)*TZ(IP7)*A3
C3150   371 CONTINUE
C3160   370 CONTINUE
C3180       Z(L2)=U*C2*Z(L2)
C3190   331 CONTINUE
C3200   330 CONTINUE
C3210       IF(NJ.EQ.0) GO TO 900
C3220C NOTE, EDGE-JUNCTION NOT INCLUDED.
C3230C EDGE-JUNCTION LOOP.
C3240       DO 43C J=1,NJ
C3250       L2=((LW+J-1)*(LC2+LE)+LC2
C3260       DO 431 IC=1,NC
C3270       DO 431 IP=1,NM
C3290C COMPUTE SUBMATRIX INDICES.
C3310       L2=L2+1
C3330       Z(L2)=0.
C3340C EDGE-JUNCTION(WIRE) IMPEDANCE ELEMENTS COMPUTED.
C3360C EDGE-JUNCTION(DISK) IMPEDANCE ELEMENTS COMPUTED.
C3380   431 CONTINUE
```

```
 430  CONTINUE
*** END CALCULATION OF Z SUBMATRIX ***
 900  CONTINUE
C PRINT OUT IMPEDANCE MATRIX Z BY SUBMATRIX BY COLUMNS.
      LI(1)=LC
      LI(2)=LC
      LI(3)=LE
      LJ(1)=LW
      LJ(2)=NJ
      CALL ZLIST(3,2,LI,LJ,NAMI,NAMJ,Z)
C WRITE Z(M) TO DISK.
      WRITE(7) (Z(I),I=1,LCWJ)
 998  CONTINUE
      STOP
      END
C-----------------------------------------------------------------
      SUBROUTINE BOTIN
C
C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
C
      COMMON /BOT1/ NMODE,NPT,NBAND
      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
      COMMON /BOT3/ DH(82),SV(82),CV(82)
      COMMON /BOT5/ T(160),TP(160),TZ(160)
      COMMON /BOT6/ IEDGE,IUNIT
      READ(5,49) NMODE,NPT,NBAND
      READ(5,49) NP
 49   FORMAT(3I3)
      WRITE(6,48) NMODE,NPT,NBAND,NP
 48   FORMAT(32H     NMODE    NPT     NBAND      NP,/,4I8)
      READ(5,53)(YB(I),I=1,NP)
      READ(5,53)(XB(I),I=1,NP)
 53   FORMAT(10F8.4)
      WRITE(6,55)
 55   FORMAT(/,3H YB)
      WRITE(6,46)(YB(I),I=1,NP)
 46   FORMAT(1X,10F8.4)
      WRITE(6,56)
 56   FORMAT(/,3H XB)
      WRITE(6,46)(XB(I),I=1,NP)
C PLOT THE BODY COORDINATES.
      CALL PLOT8(XB,YB,NP,41)
      READ(5,53) BL
      WRITE(6,47) BL
 47   FORMAT(//,21H HALF-LENGTH OF BOT =,F12.4)
C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
      IF(MOD(NP,2).NE.1) GO TO 980
C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
      IEDGE=1
      IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
      IEDGE=0
      YB(NP+1)=YB(2)
```

```
C3675       XB(NP+1)=XB(2)
C3680       YB(NP+2)=YB(3)
C3685       XB(NP+2)=XB(3)
C3690       NP=NP+2
C3695       WRITE(6,66) NP
C3700  66   FORMAT(//,39H BOT GENERATING CURVE IS CLOSED.  NP = ,I3)
C3705  10   CONTINUE
C3710C COMPUTATION OF BODY SEGMENT PARAMETERS.
C3715       DO 57 I=2,NP
C3720       I2=I-1
C3725       RR1=YB(I)-YB(I2)
C3730       RR2=XB(I)-XB(I2)
C3735       DH(I2)=SQRT(RR1*RR1+RR2*RR2)
C3740       XB1(I2)=.5*(XB(I)+XB(I2))
C3745       YB1(I2)=.5*(YB(I)+YB(I2))
C3750       SV(I2)=RR1/DH(I2)
C3755       CV(I2)=RR2/DH(I2)
C3760  57   CONTINUE
C3765C CHECK IF BOT SEGMENTATION IS UNIFORM.
C3770       IUNIF=0
C3775       NP1=NP-1
C3780       DO 60 I=2,NP1
C3785       RR1=DH(I)/DH(1)
C3790       IF(RR1.LT.0.99 .OR. RR1.GT.1.01) GO TO 20
C3795  60   CONTINUE
C3800       IUNIF=1
C3805       WRITE(6,67)
C3810  67   FORMAT(//,' BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
C3815  20   CONTINUE
C3820C COMPUTATION OF TRIANGLE FUNCTIONS T.
C3825       NM=(NP-3)/2
C3830       DO 74 J=1,NM
C3835       J2=2*(J-1)+1
C3840       J3=J2+1
C3845       J4=J3+1
C3850       J5=J4+1
C3855       J6=4*(J-1)+1
C3860       J7=J6+1
C3865       J8=J7+1
C3870       J9=J8+1
C3875       DEL1=DH(J2)+DH(J3)
C3880       DEL2=DH(J4)+DH(J5)
C3885       TP(J6)=1./DEL1
C3890       TP(J7)=1./DEL1
C3895       TP(J8)=-1./DEL2
C3900       TP(J9)=-1./DEL2
C3905       T(J6)=DH(J2)/2./DEL1
C3910       T(J7)=(DH(J2)+DH(J3)/2.)/DEL1
C3915       T(J8)=(DH(J4)/2.+DH(J5))/DEL2
C3920       T(J9)=DH(J5)/2./DEL2
C3925    74 CONTINUE
C3930       NM4=NM*4
```

```
C3935         DO 75 J=1,NM4
C3940  75     TZ(J)=T(J)
C3945         IF(IEDGE.EQ.0) GO TO 76
C3950         TZ(1)=2.-T(1)
C3955         TZ(2)=2.-T(2)
C3960         TZ(NM4-1)=2.-T(NM4-1)
C3965         TZ(NM4)=2.-T(NM4)
C3970  76     CONTINUE
C3975         RETURN
C3980  980    WRITE(6,981)
C3985  981    FORMAT(//,' **** ERROR IN BOT INPUT')
C3990         STOP
C3995         END
C4000C ********************************************************
C4005         SUBROUTINE PLOTB(X,Y,N,NR)
C4010C ********************************************************************
C4015C
C4020C WRITTEN 2/14/74   BY J. M. PUTNAM        DEPT 220      X23877
C4025C
C4030C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C4035C    N IS THE NUMBER OF POINTS TO BE PLOTTED.
C4040C    NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C4045C    NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C4050C       NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C4055C
C4060         REAL X(1),Y(1),HEAD(10)
C4065         INTEGER  LINE(101),BLANK,STAR,PLUS
C4070         DATA BLANK,STAR,PLUS /1H ,1H*,1H+/
C4075         NC=51
C4080         N10=(NC-1)/10
C4085         WRITE(6,500)
C4090  500    FORMAT(//,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
C4095         WRITE(6,504)
C4100         XMIN=X(1)
C4105         XMAX=X(1)
C4110         YMIN=Y(1)
C4115         YMAX=Y(1)
C4120         DO 6 I=1,N
C4125         IF(X(I).LT.XMIN) XMIN=X(I)
C4130         IF(X(I).GT.XMAX) XMAX=X(I)
C4135         IF(Y(I).LT.YMIN) YMIN=Y(I)
C4140         IF(Y(I).GT.YMAX) YMAX=Y(I)
C4145  6      CONTINUE
C4150         DEL=XMAX-XMIN
C4155         IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C4160         XMAX=XMIN+DEL
C4165         YMAX=YMIN+DEL
C4170         DO 5 I =1,N10
C4175         Z=I
C4180      5  HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
C4185         DY=(YMAX-YMIN)/(NR-1)
C4190         IEDGE=1
```

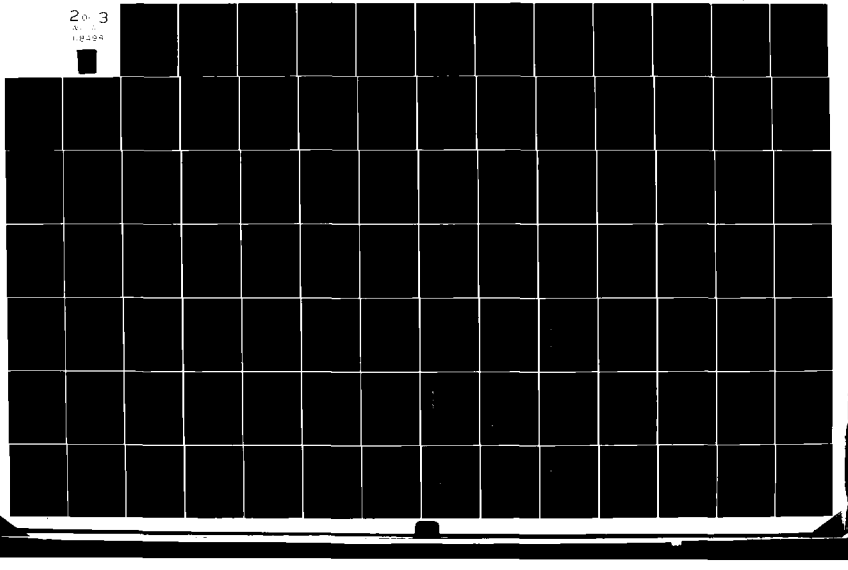
```
04195       IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
04200       Z=YMAX+DY
04205       YL=Z-DY/2.
04210       DO 7 J=1,NR
04215       DO 8 K=1,NC
04220  8    LINE(K)=BLANK
04225       Z=Z-DY
04230       YU=YL
04235       YL=Z-DY/2.
04240       DO 9 I=1,N
04245       IF(Y(I).GE.YU) GO TO 9
04250       IF(Y(I).LT.YL) GO TO 9
04255       K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
04260       IF(K.GT.NC) K=NC
04265       LINE(K)=STAR
04270       IF(MOD(I,2).EQ.1) LINE(K)=PLUS
04275       IF(IEDGE.EQ.0) GO TO 9
04280       IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
04285  9    CONTINUE
04290       WRITE(6,508) Z,(LINE(K),K=1,NC)
04295  7    CONTINUE
04300       WRITE(6,504)
04305       WRITE( 6,3002)
04310       WRITE( 6,507) XMIN,(HEAD(I),I=1,N10)
04315       RETURN
04320   504 FORMAT ( 1X, 14(1H-), 1H., 10(5H----.), 1H- )
04325   507 FORMAT(10X,11(F10.4))
04330   508 FORMAT (1X, F12.4,1X, 1HI, 51A1, 1HI )
04335  3002 FORMAT(4X,7HYH / KH,4X,1HI,5(9X,1HI))
04340       END
04345C -------------------------------------------------
04350       SUBROUTINE CAPIN
04355C
04360C READ CAP INPUTS AND COMPUTE CAP ARRAYS.
04365C
04370       COMMON /BDT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
04375       COMMON /CAP1/ NC,XC,YC,ZC(2)
04380       COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
04385       COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
04390       COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
04395       COMMON /EDG1/ NE,ZE(2),ZBE(10)
04400       COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
04405       READ(5,1) NC,NPR,NE
04410  1    FORMAT(3I3)
04415       IF(NE.NE.0) NE=NC
04420       WRITE(6,3) NC,NPR,NE
04425  3    FORMAT(16H1    NC   NPR    NE,/,3I5,//)
04430       IF(NC.EQ.0) RETURN
04435       READ(5,2) XC,YC
04440  2    FORMAT(10F8.4)
04445       READ(5,2) (ZC(I),I=1,NC)
04450       READ(5,2) (RHOC(I),I=1,NPR)
```

```
04455       IF(NE.NE.0) READ(5,2) (ZE(I),I=1,NE)
04460       WRITE(6,4)
04465  4    FORMAT(37H  CAP       XC       YC       ZC       ZE,/)
04470       DO 1CC I=1,NC
04475       IF(NE.EQ.0) WRITE(6,5) I,XC,YC,ZC(I)
04480       IF(NE.NE.0) WRITE(6,5) I,XC,YC,ZC(I),ZE(I)
04485 1CC   CONTINUE
04490  5    FORMAT(I4,4X,4F8.4)
04495       WRITE(6,6)
04500  6    FORMAT(//,5H RHOC)
04505       WRITE(6,7) (RHOC(I),I=1,NPR)
04510  7    FORMAT(1X,10F8.4)
04515       IF(MOD(NPR,2).NE.1) GO TO 980
04520       DO 120 I=2,NPR
04525 120   IF(RHOC(I).LE.RHOC(I-1)) GO TO 980
04530C COMPUTATION OF CAP SECTOR PARAMETERS.
04535       DO 47 I=1,NP
04540       RC(I)=SQRT((YB(I)-YC)**2+(XB(I)-XC)**2)
04545 47    CONTINUE
04550       DO 57 I=2,NP
04555       I2=I-1
04560       RR1=YB1(I2)-YC
04565       RR2=XB1(I2)-XC
04570       RC1(I2)=SQRT(RR1*RR1+RR2*RR2)
04575       AC(I2)=ABS((XB(I)-XC)*(YB(I)+YC)+(XB(I2)-XB(I))*
04580      1  (YB(I2)+YB(I))+(XC-XB(I2))*(YC+YB(I2)))/2.
04585       SPC(I2)=RR1/RC1(I2)
04590       CPC(I2)=RR2/RC1(I2)
04595 57    CONTINUE
04600       DO 67 I=2,NPR
04605       I2=I-1
04610       RHOC1(I2)=(RHOC(I)+RHOC(I2))/2.
04615       DRHOC(I2)=RHOC(I)-RHOC(I2)
04620 67    CONTINUE
04625C COMPUTATION OF CAP TRIANGLE FUNCTIONS.
04630       LC=(NPR-3)/2
04635       DO 74 J=1,LC
04640       J2=2*(J-1)+1
04645       J3=J2+1
04650       J4=J3+1
04655       J5=J4+1
04660       J6=4*(J-1)+1
04665       J7=J6+1
04670       J8=J7+1
04675       J9=J8+1
04680       DEL1=DRHOC(J2)+DRHOC(J3)
04685       DEL2=DRHOC(J4)+DRHOC(J5)
04690       TPCR(J6)=1./DEL1
04695       TPCR(J7)=1./DEL1
04700       TPCR(J8)=-1./DEL2
04705       TPCR(J9)=-1./DEL2
04710       TCR(J6)=DRHOC(J2)/2./DEL1
```

```
04715       TCR(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
04720       TCR(J8)=(DRHOC(J4)/2.+DRHOC(J5))/DEL2
04725       TCR(J9)=DRHOC(J5)/2./DEL2
04730    74 CONTINUE
04735       LC4=LC*4
04740       DO 75 I=1,LC4
04745       TCT(I)=TCR(I)
04750       TPCT(I)=TPCR(I)
04755    75 CONTINUE
04760       TCT(LC4-1)=2.-TCT(LC4-1)
04765       TCT(LC4)=2.-TCT(LC4)
04770       TPCT(LC4-1)=-TPCT(LC4-1)
04775       TPCT(LC4)=-TPCT(LC4)
04780       IF(RHOC(1).EQ.0.0) GO TO 76
04785       TCT(1)=2.-TCT(1)
04790       TCT(2)=2.-TCT(2)
04795       TPCT(1)=-TPCT(1)
04800       TPCT(2)=-TPCT(2)
04805    76 CONTINUE
04810       IF(NE.EQ.0) RETURN
04815C COMPUTATION OF EDGE HALF TRIANGLE FUNCTIONS.
04820       DO 80 IC=1,NC
04825       J2=NPR-2
04830       J3=J2+1
04835       J6=2*(IC-1)+1
04840       J7=J6+1
04845       DEL1=DRHOC(J2)+DRHOC(J3)
04850       TPCE(J6)=1./DEL1
04855       TPCE(J7)=1./DEL1
04860       TCE(J6)=DRHOC(J2)/2./DEL1
04865       TCE(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
04870       DEL2=ZE(IC)-ZC(IC)
04875       ZBE(J6)=ZC(IC)+0.25*DEL2
04880       ZBE(J7)=ZC(IC)+0.75*DEL2
04885       IF(DEL2.LT.0.0) GO TO 78
04890C EDGE IS AT Z=-L.
04895       TPBE(J6)=-1./DEL2
04900       TPBE(J7)=-1./DEL2
04905       TBE(J6)=0.75
04910       TBE(J7)=0.25
04915       GO TO 80
04920    78 CONTINUE
04925C EDGE IS AT Z=+L.
04930       TPBE(J6)=1./DEL2
04935       TPBE(J7)=1./DEL2
04940       TBE(J6)=-0.75
04945       TBE(J7)=-0.25
04950    80 CONTINUE
04955       RETURN
04960   986 WRITE(6,981)
04965   981 FORMAT(//,' **** ERROR IN CAP INPUT')
04970       STOP
```

```
C4975       END
C4980C ------------------------------------------------------
C4985       SUBROUTINE WIREIN
C4990C
C4995C READ WIRE COORDINATES AND COMPUTE WIRE SEGMENT ARRAYS.
C5000C
C5005       COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
C5010      1 XW1(100),YW1(100),ZW1(100)
C5015       COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
C5020       COMMON /WIRE3/ NW,INDW(6),RADW(5)
C5025       COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
C5030       COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
C5035       COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
C5040       COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
C5045       COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
C5050       COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
C5055       COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
C5060       NW=0
C5065       NJ=0
C5070       LW=0
C5075       READ(5,51) NW,NPW,NJ
C5080  51    FORMAT(3I3)
C5085       WRITE(6,1) NW,NPW,NJ
C5090  1     FORMAT(24H1      NW      NPW      NJ,/,3I8)
C5095       IF(NW.EQ.0) RETURN
C5100       READ(5,53) (XW(I),I=1,NPW)
C5105       READ(5,53) (YW(I),I=1,NPW)
C5110       READ(5,53) (ZW(I),I=1,NPW)
C5115  53    FORMAT(10F8.4)
C5120       READ(5,52) (INDW(I),I=1,NW)
C5125  52    FORMAT(10I8)
C5130       INDW(NW+1)=NPW+1
C5135       READ(5,53) (RADW(I),I=1,NW)
C5140       IF(NJ.EQ.0) GO TO 5C
C5145C NOTE, INDJW MUST BE MONOTONIC INCREASING.
C5150       READ(5,52) (INDJW(I),I=1,NJ)
C5155       READ(5,53) (RADD(I),I=1,NJ)
C5160       READ(5,53) (UXJ(I),I=1,NJ)
C5165       READ(5,53) (UYJ(I),I=1,NJ)
C5170       READ(5,53) (UZJ(I),I=1,NJ)
C5175  5C    IERW=C
C5180       IERJ=0
C5185       WRITE(6,61)
C5190  61    FORMAT(/,22X,'WIRE COORDINATES',2CX,'JUNCTION PARAMETERS',/,
C5195      1 26X,'IW',4X,'XW',6X,'YW',6X,'ZW',
C5200      2 1CX,'IJ',4X,'RADD',5X,'UXJ',5X,'UYJ',5X,'UZJ')
C5205       IJ=1
C5210C THIS LOOP LISTS WIRE/JUNCTION POINTS, WHILE CHECKING THE FOLLOWING:
C5215C   1) EACH WIRE MUST CONTAIN AN ODD NUMBER OF POINTS.
C5220C   2) EACH JUNCTION MUST EITHER START OR TERMINATE A WIRE.
C5225C   3) CHECK THAT ALL JUNCTION POINTS ARE FOUND.
C5230       DO 1CC IW=1,NW
```

AD-A118 498 MCDONNELL DOUGLAS RESEARCH LABS ST LOUIS MO F/G 9/2
ALGORITHM FOR SURFACE OF TRANSLATION ATTACHED RADIATORS (A-STAR--ETC(U)
MAY 82 L N MEDGYESI-MITSCHANG, J M PUTNAM F30602-80-C-0106
UNCLASSIFIED RADC-TR-82-113-VOL-3 NL



```
C5235         WRITE(6,62) IW,RADW(IW)
C5240  62     FORMAT(2X,'WIRE',I3,' RADW=',F8.4)
C5245         I1=INDW(IW)
C5250         I2=INDW(IW+1)-1
C5255C CHECK FOR AN ODD NUMBER OF POINTS ON WIRE IW.
C5260         IF(MOD(I2-I1+1,2).NE.1) IERW=1
C5265         DO 9C I=I1,I2
C5270         WRITE(6,63) I,XW(I),YW(I),ZW(I)
C5275  63     FORMAT(25X,I3,3F8.4)
C5280C CHECK IF WIRE POINT I IS A JUNCTION POINT.
C5285         IF(IJ.GT.NJ) GO TO 90
C5290         IF(INDJW(IJ).NE.I) GO TO 90
C5295C CHECK THAT JUNCTION POINT IJ IS AT THE START OR END OF WIRE.
C5300         IF(I.NE.I1 .AND. I.NE.I2) IERJ=1
C5305         WRITE(6,64) IJ,RADJ(IJ),UXJ(IJ),UYJ(IJ),UZJ(IJ)
C5310  64     FORMAT('+',58X,I3,4F8.4)
C5315C DETERMINE DIRECTION IN WHICH WIRE LEAVES JUNCTION POINT.
C5320         INDTJ(IJ)=1
C5325         IF(I.EQ.I2) INDTJ(IJ)=-1
C5330C COMPUTE JUNCTION PARAMETERS.
C5335         RADJ(IJ)=RADW(IW)
C5340         XJ(IJ)=XW(I)
C5345         YJ(IJ)=YW(I)
C5350         ZJ(IJ)=ZW(I)
C5355         IJ=IJ+1
C5360  90     CONTINUE
C5365  100    CONTINUE
C5370C CHECK FOR WIRE OR JUNCTION INPUT ERRORS.
C5375         IF(IERW.NE.C) GO TO 980
C5380         IF(IJ-1.NE.NJ .OR. IERJ.NE.0) GO TO 990
C5385C COMPUTATION OF WIRE SEGMENT PARAMETERS.
C5390         DO 57 I=2,NPW
C5395         I2=I-1
C5400         DXW(I2)=XW(I)-XW(I2)
C5405         DYW(I2)=YW(I)-YW(I2)
C5410         DZW(I2)=ZW(I)-ZW(I2)
C5415         DHW(I2)=SQRT(DXW(I2)**2+DYW(I2)**2+DZW(I2)**2)
C5420         XW1(I2)=0.5*(XW(I)+XW(I2))
C5425         YW1(I2)=C.5*(YW(I)+YW(I2))
C5430         ZW1(I2)=0.5*(ZW(I)+ZW(I2))
C5435   57    CONTINUE
C5440C COMPUTATION OF WIRE TRIANGLE FUNCTIONS TW.
C5445         LW=0
C5450         DO 75 IW=1,NW
C5455         I1=INDW(IW)
C5460         I2=INDW(IW+1)-1
C5465         LW1=(I2-I1-2)/2
C5470         DO 74 J=1,LW1
C5475         LW=LW+1
C5480         J2=2*(J-1)+I1
C5485         J3=J2+1
C5490         J4=J3+1
```

```
C5495         J5=J4+1
C55CC         J6=4*(LW-1)+1
C55C5         J7=J6+1
C551C         J8=J7+1
C5515         J9=J8+1
C5520         INDTW(LW)=J2
C5525         DEL1=DHW(J2)+DHW(J3)
C553C         DEL2=DHW(J4)+DHW(J5)
C5535         TPW(J6)=1./DEL1
C5540         TPW(J7)=1./DELI
C5545         TPW(J8)=-1./DEL2
C555C         TPW(J9)=-1./DEL2
C5555         TW(J6)=DHW(J2)/2./DEL1
C556C         TW(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C5565         TW(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C5570         TW(J9)=DHW(J5)/2./DEL2
C5575  74     CONTINUE
C5580  75     CONTINUE
C5585C NOTE, LW=(NPW-NW)/2-NW
C5590         IF(NJ.EQ.0) RETURN
C5595C COMPUTATION OF JUNCTION HALF TRIANGLE FUNCTIONS TJ.
C5600         DO 85 IJ=1,NJ
C5605         IF(INDTJ(IJ).GT.0) GO TO 80
C5610C JUNCTION IS AT THE END OF A WIRE.
C5615         J2=INDJW(IJ)-2
C5620         J3=J2+1
C5625         J6=2*(IJ-1)+1
C5630         J7=J6+1
C5635         INDTJ(IJ)=J2
C5640         DEL1=-(DHW(J2)-DHW(J3)
C5645         TPJ(J6)=1./DEL1
C5650         TPJ(J7)=1./DEL1
C5655         TJ(J6)=DHW(J2)/2./DEL1
C5660         TJ(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C5665         GO TO 85
C5670C JUNCTION IS AT THE START OF A WIRE.
C5675  80     J4=INDJW(IJ)
C5680         J5=J4+1
C5685         J8=2*(IJ-1)+1
C5690         J9=J8+1
C5695         INDTJ(IJ)=J4
C5700         DEL2=DHW(J4)+DHW(J5)
C5705         TPJ(J8)=-1./DEL2
C5710         TPJ(J9)=-1./DEL2
C5715         TJ(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C5720         TJ(J9)=DHW(J5)/2./DEL2
C5725  85     CONTINUE
C5730C COMPUTE UNIT VECTORS FOR JUNCTION(DISK).
C5735         DO 95 IJ=1,NJ
C5740C NORMAL UNIT VECTOR (UJ).
C5745         RR=SQRT(UXJ(IJ)**2+UYJ(IJ)**2+UZJ(IJ)**2)
C5750         UXJ(IJ)=UXJ(IJ)/RR
```

```
05755       UYJ(IJ)=UYJ(IJ)/RR
05760       UZJ(IJ)=UZJ(IJ)/RR
05765C FIND 2 ORTHOGONAL UNIT VECTORS IN THE PLANE OF THE DISK (UJ1 & UJ2).
05770       UXJ1(IJ)=0.0
05775       UYJ1(IJ)=0.0
05780       UZJ1(IJ)=0.0
05785       UXJ2(IJ)=0.0
05790       UYJ2(IJ)=0.0
05795       UZJ2(IJ)=0.0
05800       IF(UXJ(IJ).EQ.0.0) GO TO 91
05805C FIND INTERSECTION WITH X-Y PLANE.
05810       UYJ1(IJ)=1.0
05815       UXJ1(IJ)=-UYJ(IJ)/UXJ(IJ)
05820C FIND INTERSECTION WITH X-Z PLANE.
05825       UZJ2(IJ)=1.0
05830       UXJ2(IJ)=-UZJ(IJ)/UXJ(IJ)
05835       GO TO 94
05840 91    IF(UYJ(IJ).EQ.0.0) GO TO 92
05845C FIND INTERSECTION WITH Y-Z PLANE.
05850       UZJ1(IJ)=1.0
05855       UYJ1(IJ)=-UZJ(IJ)/UYJ(IJ)
05860C FIND INTERSECTION WITH X-Y PLANE.
05865       UXJ2(IJ)=1.0
05870       UYJ2(IJ)=-UXJ(IJ)/UYJ(IJ)
05875       GO TO 94
05880 92    IF(UZJ(IJ).EQ.0.0) GO TO 94
05885C FIND INTERSECTION WITH X-Z PLANE.
05890       UXJ1(IJ)=1.0
05895       UZJ1(IJ)=-UXJ(IJ)/UZJ(IJ)
05900C FIND INTERSECTION WITH Y-Z PLANE.
05905       UYJ2(IJ)=1.0
05910       UZJ2(IJ)=-UYJ(IJ)/UZJ(IJ)
05915 94    CONTINUE
05920       RR=SQRT(UXJ1(IJ)**2+UYJ1(IJ)**2+UZJ1(IJ)**2)
05925       UXJ1(IJ)=UXJ1(IJ)/RR
05930       UYJ1(IJ)=UYJ1(IJ)/RR
05935       UZJ1(IJ)=UZJ1(IJ)/RR
05940       RR=SQRT(UXJ2(IJ)**2+UYJ2(IJ)**2+UZJ2(IJ)**2)
05945       UXJ2(IJ)=UXJ2(IJ)/RR
05950       UYJ2(IJ)=UYJ2(IJ)/RR
05955       UZJ2(IJ)=UZJ2(IJ)/RR
05960 95    CONTINUE
05965       RETURN
05970 980   WRITE(6,981)
05975 981   FORMAT(//,' **** ERROR IN WIRE INPUT')
05980       STOP
05985 990   WRITE(6,991)
05990 991   FORMAT(//,' **** ERROR IN JUNCTION INPUT')
05995       STOP
06000       END
08550C ------------------------------------------
08555       SUBROUTINE ZLIST(NI,NJ,LI,LJ,NAMI,NAMJ,Z)
```

```
08560C WRITE Z TO THE LINE-PRINTER BY SUBMATRIX BY COLUMNS.
08565       COMPLEX Z(1)
08570       DIMENSION LI(1),LJ(1),NAMI(1),NAMJ(1)
08575       NCOL=0
08580       DO 100 I=1,NI
08585 100   NCOL=NCOL+LI(I)
08590       NR=1
08595       DO 300 J=1,NJ
08600       IF(LJ(J).EQ.0) GO TO 300
08605       NC=1
08610       DO 200 I=1,NI
08615       IF(LI(I).EQ.0) GO TO 200
08620       WRITE(6,1) NAMI(I),NAMJ(J)
08625 1     FORMAT(/,2H Z,1X,A4,3H - ,A4)
08630       KI=LI(I)
08635       KJ=LJ(J)
08640       K1=(NR-1)*NCOL+NC
08645       DO 150 KK=1,KJ
08650       K2=K1+KI-1
08655       WRITE(6,2) (Z(K),K=K1,K2)
08660 2     FORMAT(1X,10G11.4)
08665 150   K1=K1+NCOL
08670       NC=NC+LI(I)
08675 200   CONTINUE
08680       NR=NR+LJ(J)
08685 300   CONTINUE
08740       RETURN
08750       END
08760/EOR
```

```
BBBBBBBBB      OOOOOOOO   TTTTTTTTTTT    IIIIIIII   NN         NN  VV         VV
BBBBBBBBBB    OOOOOOOOOO  TTTTTTTTTTT    IIIIIIII   NN         NN  VV         VV
BB        BB  OO      OO      TT           II      NNN        NN   VV       VV
BB        BB  OO      OO      TT           II      NNNN       NN    VV     VV
BB        BB  OO      OO      TT           II      NNNNN      NN     VV   VV
BB        BB  OO      OO      TT           II      NN NN      NN      VV VV
BBBBBBBBBB    OO      OO      TT           II      NN  NN     NN      VV VV
BBBBBBBBB     OO      OO      TT           II      NN   NN    NN       VVV
BB        BB  OO      OO      TT           II      NN    NN   NN       VVV
BB        BB  OO      OO      TT           II      NN     NN  NN       VVV
BB        BB  OO      OO      TT           II      NN      NNNNN      VV VV
BB        BB  OO      OO      TT           II      NN       NNNN      VV VV
BB        BB  OO      OO      TT           II      NN       NNNN     VV   VV
BB        BB  OOOO  OOOO      TT           II      NN        NNN     VVVVV
BBBBBBBBBB     OOOOOOOO       TT       IIIIIIII    NN        NN      VVVVV
BBBBBBBBB       OOOOOO        TT       IIIIIIII    NN        NN       VVV
```

```
C01COBOTI,P3C,T1CO,CM145CCO,STCX3.
C0110ACCOUNT,MG326S,BOT2A.
C0120BANNERS(OUTPUT) *J. PUTNAM*DEPT 220*BLD 110-4**
C0130ATTACH(INFIL=ZSSX)
00135DEFINE(OUTFIL=YSSX)
C014CFTN(R=C,OPT=2)
C0170LDSET(PRESET=INDEF)
00180LGO.
CC220/EOR
C0230       PROGRAM BOTINV(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C0231      1 INFIL,TAPE1=INFIL,OUTFIL,TAPE7=OUTFIL)
C0232C
C0233C BOTINV INVERTS THE BOT Z MATRIX, YIELDING THE BOT Y MATRIX.
C0234C
C0235C UNIT 5 IS THE CARD READER.
C0236C UNIT 6 IS THE LINE PRINTER.
C0237C UNIT 1 IS A DISK FILE CONTAINING THE Z MATRIX.
00238C UNIT 7 IS A DISK FILE FOR OUTPUT OF THE Y MATRIX.
00239C
C0250       COMPLEX Z(12544),ZI(1792),LOAD(16)
C0255       DIMENSION WGHT(16)
00260       COMPLEX Z1,Z2,Z3,Z4
C0265       COMPLEX Z1T,Z2T,Z3T,Z4T
CC270       DIMENSION NZ(31)
C0280       COMMON NM,JK(4),LR(112)
C0285       NFOUND=0
C0290       READ(5,1) NMODE,NBAND
00291 1     FORMAT(2I3)
C0292       READ(1) NP
C0293       NBAND1=NBAND-1
00305       WRITE(6,6) NP,NMODE,NBAND
C0310 6     FORMAT(/,32H          NP        NMODE      NBAND,/,3I10)
C0326       LS=NP-3
00330       NM=LS/2
C0340       LSS=LS*LS
00342C READ SURFACE IMPEDANCE LOADING.
C0345       READ(5,8) (LOAD(I),I=1,NM)
C0346 8     FORMAT(10F8.4)
00347       NM1=NM+1
C0348       READ(5,8) (LOAD(I),I=NM1,LS)
CC349       WRITE(6,9)
00350 9     FORMAT(/,'  T AND Z SURFACE IMPEDANCE LOADING (COMPLEX)')
C0351       WRITE(6,59) (LOAD(I),I=1,NM)
C0352 59    FORMAT(//,(1X,10F8.4))
CC353       WRITE(6,59) (LOAD(I),I=NM1,LS)
00354       READ(1) (WGHT(I),I=1,NM)
C0355       READ(1) (WGHT(I),I=NM1,LS)
C0356       DO 6CC I=1,LS
C0357 600   LOAD(I)=LOAD(I)*WGHT(I)
CC362       I3=NM*LS
C0364       I4=I3+NM
00370       JK(1)=1
```

```
00380       JK(2)=NM+1
00390       JK(3)=2*NM*NM+1
00395       JK(4)=JK(3)+NM
00398       WRITE(7) NMODE,NP,0,0,0,0,0,0
C0400       WRITE(7) 0,0
C0401       WRITE(6,2)
C0402  2    FORMAT(//,46H  THE MINIMUM ROTINV ARRAY DIMENSIONS FOR THIS,
00403      1 24H PROBLEM ARE AS FOLLOWS:,//,
00404      2 49H          Z          ZI     LOADEWGHT      NZ         LR,/)
C0405       IF(NBAND.LE.0) STOP
00410       IF(NBAND.EQ.1) GO TO 40C
C0420       IF(NBAND.LT.2*NMODE-1) GO TO 200
00430C *****  INVERT THE ENTIRE Z ROT MATRIX  *****
C0440       LL=LS*(2*NMODE-1)
00450       L3=NM*LL
00460       L4=L3+NM
C0461       K1=LL*LL
C0462       K2=LSS
C0463       K3=LS
C0464       K4=0
C0465       K5=LL
C0466       WRITE(6,3) K1,K2,K3,K4,K5
C0467  3    FORMAT(5I10)
C0468       WRITE(6,53)
00470  53   FORMAT(//,22H FULL MATRIX INVERSION)
C0472       NEXP=NMODE*NMODE
00474  5    READ(1) M,N
C0480       IF(EOF(1)) 100,10
C0490  10   READ(1) (ZI(I),I=1,LSS)
C0500       IF(IABS(M).GE.NMODE .OR. IABS(N).GE.NMODE) GO TO 5
C0505       NFOUND=NFOUND+1
C0506C ADD  SURFACE IMPEDANCE TO DIAGONAL.
C0507       I=1
00508       DO 11 J=1,LS
00510       IF(J.GT.NM .AND. M*N.EQ.0) GO TO 11
C0511       ZI(I)=ZI(I)+LOAD(J)
C0512  11   I=I+LS+1
C0515       K1=(N+NMODE-1)*LL*LS+(M+NMODE-1)*LS
C0520       K2=(M+NMODE-1)*LL*LS+(N+NMODE-1)*LS
C0530       K3=(-N+NMODE-1)*LL*LS+(-M+NMODE-1)*LS
C0540       K4=(-M+NMODE-1)*LL*LS+(-N+NMODE-1)*LS
00550C INSERT ZI INTO THE Z MATRIX, USING THE SYMMETRIES.
00560       DO 20 J=1,NM
C0570       II=(J-1)*LS
00575       IIT=J-LS
C0580       DO 15 I=1,NM
C0590       II=II+1
00595       IIT=IIT+LS
00600       K1=K1+1
C0610       K2=K2+1
C0620       K3=K3+1
C0630       K4=K4+1
```

```
00635C ZTT , ZZT , ZTZ ,AND ZZZ MATRIX ELEMENTS.
00640          Z1=ZI(II)
00650          Z2=ZI(II+NM)
00660          Z3=ZI(II+I3)
00670          Z4=ZI(II+I4)
00671C TRANSPOSE ELEMENTS.
00672          Z1T=ZI(IIT)
00674          Z2T=ZI(IIT+NM)
00676          Z3T=ZI(IIT+I3)
00678          Z4T=ZI(IIT+I4)
00680C Z(M,N).
00690          Z(K1)=Z1
00700          Z(K1+NM)=Z2
00710          Z(K1+L3)=Z3
00720          Z(K1+L4)=Z4
00730          IF(M.EQ.N) GO TO 12
00740C Z(N,M).
00750          Z(K2)=Z1T
00760          Z(K2+NM)=-Z3T
00770          Z(K2+L3)=-Z2T
00780          Z(K2+L4)=Z4T
00790          IF(M.EQ.-N) GO TO 15
00800C Z(-M,-N).
00810          Z(K3)=Z1
00820          Z(K3+NM)=-Z2
00830          Z(K3+L3)=-Z3
00840          Z(K3+L4)=Z4
00850  12      IF(M.EQ.-N) GO TO 15
00860C Z(-N,-M).
00870          Z(K4)=Z1T
00880          Z(K4+NM)=Z3T
00890          Z(K4+L3)=Z2T
00900          Z(K4+L4)=Z4T
00910  15      CONTINUE
00920          K1=K1+LL-NM
00930          K2=K2+LL-NM
00940          K3=K3+LL-NM
00950          K4=K4+LL-NM
00960  20      CONTINUE
00970          GO TO 5
00980  100     CONTINUE
00985          IF(NFOUND.NE.NEXP) GO TO 490
00987C INVERT THE Z MATRIX.
00990          CALL LINEQ(LL,Z,LR)
01000C OUTPUT Z MATRIX BY MODE NUMBERS.
01010          KMODE=2*NMODE-1
01020          DO 150 NN=1,KMODE
01030          N=-NMODE+NN
01040          DO 150 MM=1,KMODE
01050          M=-NMODE+MM
01060          II=0
01070          K=(N+NMODE-1)*LL*LS+(M+NMODE-1)*LS
```

```
01080         DO 125 J=1,LS
01090         DO 120 I=1,LS
01100         II=II+1
01110         K=K+1
01120         ZI(II)=Z(K)
01130  120    CONTINUE
01140         K=K+LL-LS
01150  125    CONTINUE
01160         CALL LIST(M,N,ZI)
01170  150    CONTINUE
01180         GO TO 500
01190C *****   INVERT THE BANDED Z BOT MATRIX  *****
01200  200    KMODE=2*NMODE-1
01210         K1=NBAND
01220         K2=KMODE-NBAND+2
01230         KZ=KMODE-K1+1
01240         NZ(1)=0
01250         DO 202 I=2,KMODE
01260         I1=I-1
01270         IF(I.LE.K1) KZ=KZ-1
01280         IF(I.GT.K2) KZ=KZ+1
01290  202    NZ(I)=NZ(I1)+KZ
01291         K1=LSS*(KMODE*(2*NBAND-1)-NBAND*(NBAND-1))
01292         K2=KMODE*LSS
01293         K3=LS
01294         K4=KMODE
01295         K5=LS
01296         WRITE(6,3) K1,K2,K3,K4,K5
01298         WRITE(6,54)
01300  54     FORMAT(//,24H BANDED MATRIX INVERSION)
01302         NEXP=NMODE*NMODE-MOD(NBAND1,2)*(NMODE-(NBAND1+1)/2)**2
01304        1 +(MOD(NBAND1,2)-1)*(NMODE-NBAND1/2-1)*(NMODE-NBAND1/2)
01306  205    READ(1) M,N
01310         IF(EOF(1)) 300,210
01320  210    IF(IABS(M-N).GE.NBAND) GO TO 300
01330         READ(1) (ZI(I),I=1,LSS)
01340         IF(IABS(M).GE.NMODE .OR. IABS(N).GE.NMODE) GO TO 205
01345         NFOUND=NFOUND+1
01346C ADD SURFACE IMPEDANCE TO DIAGONAL.
01347         I=1
01348         DO 211 J=1,LS
01350         IF(J.GT.NM .AND. M*N.EQ.0) GO TO 211
01351         ZI(I)=ZI(I)+LOAD(J)
01352  211    I=I+LS+1
01355         K1=((N+NMODE-1)*KMODE+(M+NMODE-1)-NZ(N+NMODE))*LSS
01360         K2=((M+NMODE-1)*KMODE+(N+NMODE-1)-NZ(M+NMODE))*LSS
01370         K3=((-N+NMODE-1)*KMODE+(-M+NMODE-1)-NZ(-N+NMODE))*LSS
01380         K4=((-M+NMODE-1)*KMODE+(-N+NMODE-1)-NZ(-M+NMODE))*LSS
01385C INSERT ZI INTO THE Z MATPIX, USING THE SYMMETRIES.
01390         DO 220 J=1,NM
01400         II=(J-1)*LS
01405         III=J-LS
```

```
01410       DO 215 I=1,NM
01420       II=II+1
01425       IIT=IIT+LS
01430       K1=K1+1
01440       K2=K2+1
01450       K3=K3+1
01460       K4=K4+1
01465C ZTT , ZZT , ZTZ ,AND ZZZ MATRIX ELEMENTS.
01470       Z1=ZI(II)
01480       Z2=ZI(II+NM)
01490       Z3=ZI(II+I3)
01500       Z4=ZI(II+I4)
01501C TRANSPOSE ELEMENTS.
01502       Z1T=ZI(IIT)
01504       Z2T=ZI(IIT+NM)
01506       Z3T=ZI(IIT+I3)
01508       Z4T=ZI(IIT+I4)
01510C Z(M,N).
01520       Z(K1)=Z1
01530       Z(K1+NM)=Z2
01540       Z(K1+I3)=Z3
01550       Z(K1+I4)=Z4
01560       IF(M.EQ.N) GO TO 212
01570C Z(N,M).
01580       Z(K2)=Z1T
01590       Z(K2+NM)=-Z3T
01600       Z(K2+I3)=-Z2T
01610       Z(K2+I4)=Z4T
01620       IF(M.EQ.-N) GO TO 215
01630C Z(-M,-N).
01640       Z(K3)=Z1
01650       Z(K3+NM)=-Z2
01660       Z(K3+I3)=-Z3
01670       Z(K3+I4)=Z4
01680 212   IF(M.EQ.-N) GO TO 215
01690C Z(-N,-M).
01700       Z(K4)=Z1T
01710       Z(K4+NM)=Z3T
01720       Z(K4+I3)=Z2T
01730       Z(K4+I4)=Z4T
01740 215   CONTINUE
01750       K1=K1+NM
01760       K2=K2+NM
01770       K3=K3+NM
01780       K4=K4+NM
01790 220   CONTINUE
01800       GO TO 205
01810 300   CONTINUE
01815       IF(NFOUND.NE.NEXP) GO TO 490
01817C INVERT THE Z MATRIX AND THEN LIST THE INVERTED MATRIX.
01820       CALL INVBAN(LS,NMODE,NBAND,NZ,Z,ZI,LOAD)
01830       GO TO 500
```

```
01840C ***** INVERT THE DIAGONAL Z BOT MATRIX  *****
C1851  400   K1=0
01852        K2=LSS
01853        K3=LS
01854        K4=0
C1855        K5=LS
01856        WRITE(6,3) K1,K2,K3,K4,K5
01858        WRITE(6,55)
01860  55    FORMAT(//,26H DIAGONAL MATRIX INVERSION)
01862        NEXP=NMODE
01864  405   READ(1)  M,N
01870        IF(EOF(1)) 500,410
01880  410   IF(IABS(M-N).GT.0) GO TO 500
01890        IF(M.GE.NMODE) GO TO 500
01900        READ(1) (ZI(I),I=1,LSS)
C1905        NFOUND=NFOUND+1
C19C6C ADD SURFACE IMPEDANCE TO DIAGONAL.
C1907        I=1
C1908        DO 411 J=1,LS
01910        IF(J.GT.NM .AND. M*N.EQ.0) GO TO 411
C1911        ZI(I)=ZI(I)+LOAD(J)
C1912  411   I=I+LS+1
02105        CALL LINEQ(LS,ZI,LR)
02110        CALL LIST(M,N,ZI)
02120        IF(M.EQ.0)  GO TO 405
02122C COMPUTE Z(-M,-M) INVERSE FROM Z(M,M) INVERSE.
C2124C NOTE, THE OFF DIAGONAL SUBMATRICES CHANGE SIGNS.
C2130        DO 430 J=1,NM
C2131        II=(J-1)*LS
C2132        DO 425 I=1,NM
02133        II=II+1
02134        ZI(II+NM)=-ZI(II+NM)
C2135        ZI(II+I3)=-ZI(II+I3)
02136  425   CONTINUE
C2137  430   CONTINUE
02140        CALL LIST(-M,-N,ZI)
02150        GO TO 405
02152  490   WRITE(6,7)
C2154  7     FORMAT(//,41H ERROR IN THE NUMBER OF SUBMATRICES FOUND)
C2160  500   CONTINUE
02165        WRITE(6,4) NEXP,NFOUND
02166  4     FORMAT(//,33H NUMBER OF SUBMATRICES EXPECTED =,I5,//,
02167       1  30H NUMBER OF SUBMATRICES FOUND =,I8)
02170        STOP
02180        END
02185C ************************************************************
02190        SUBROUTINE LINEQ(LL,C,LR)
02195C ************************************************************
02197C COMPLEX MATRIX INVERSION ROUTINE.
02200        COMPLEX C(1),STOR,STO,ST,S
02210        DIMENSION LR(1)
02220        DO 2C I=1,LL
```

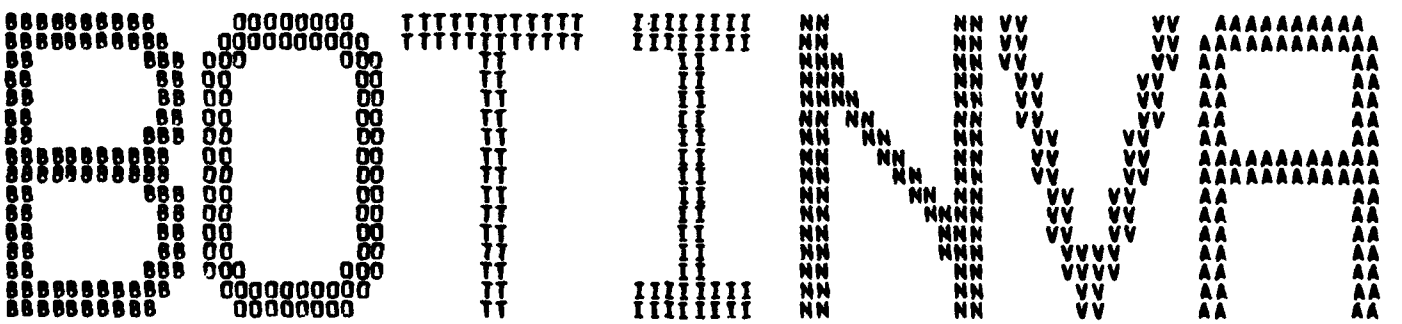
```
C2230          LR(I)=I
C2240       20 CONTINUE
C2250          M1=0
C2260          DO 18 M=1,LL
C2270          K=M
C2280          K2=M1+K
C2290          S1=CABS(C(K2))
C2300          DO 2 I=M,LL
C2310          K1=M1+I
C2320          S2=CABS(C(K1))
C2330          IF(S2-S1) 2,2,6
C2340        6 K=I
C2350          S1=S2
C2360        2 CONTINUE
C2370          LS=LR(M)
C2380          LR(M)=LR(K)
C2390          LR(K)=LS
C2400          K2=M1+K
C2410          STOR=C(K2)
C2420          J1=0
C2430          DO 7 J=1,LL
C2440          K1=J1+K
C2450          K2=J1+M
C2460          STO=C(K1)
C2470          C(K1)=C(K2)
C2480          C(K2)=STO/STOR
C2490          J1=J1+LL
C2500        7 CONTINUE
C2510          K1=M1+M
C2520          C(K1)=1./STOR
C2530          DO 11 I=1,LL
C2540          IF(I-M) 12,11,12
C2550       12 K1=M1+I
C2560          ST=C(K1)
C2570          C(K1)=0.
C2580          J1=0
C2590          DO 10 J=1,LL
C2600          K1=J1+I
C2610          K2=J1+M
C2620          C(K1)=C(K1)-C(K2)*ST
C2630          J1=J1+LL
C2640       10 CONTINUE
C2650       11 CONTINUE
C2660          M1=M1+LL
C2670       18 CONTINUE
C2680          J1=0
C2690          DO 9 J=1,LL
C2700          IF(J-LR(J)) 14,9,14
C2710       14 LRJ=LR(J)
C2720          J2=(LRJ-1)*LL
C2730       21 DO 13 I=1,LL
C2740          K2=J2+I
```

```
C2750       K1=J1+I
C2760       S=C(K2)
C2770       C(K2)=C(K1)
C2780       C(K1)=S
C2790    13 CONTINUE
C2800       LR(J)=LR(LRJ)
C2810       LR(LRJ)=LRJ
C2820       IF(J-LR(J)) 14,8,14
C2830     8 J1=J1+LL
C2840     9 CONTINUE
C2850       RETURN
C2860       END
C2865C ***************************************************************
C2870       SUBROUTINE LIST(M,N,ZI)
C2875C ***************************************************************
C2876C PRINT Z(M,N) ON THE LINE PRINTER AND WRITE TO DISK FILE.
C2880       COMPLEX ZI(1)
C2890       COMMON NM,JK(4)
C2900       LS=2*NM
C2910       LSS=LS*LS
C2915       IF(M.NE.N) GO TO 200
C2920       WRITE(6,2) M,N
C2930 2     FORMAT(4H1 M=,I3,4H  N=,I3,/)
C2940       DO 140 J=1,4
C2950       K1=JK(J)
C2960       WRITE(6,3) J
C2970 3     FORMAT(2H Y,I1)
C2980       DO 140 I=1,NM
C2990       K2=K1+NM-1
C3000       WRITE(6,4) (ZI(K),K=K1,K2)
C3010 4     FORMAT(1X,10G11.4)
C3020       K1=K1+LS
C3030 140   CONTINUE
C3035 200   CONTINUE
C3040       WRITE(7) M,N
C3050       WRITE(7) (ZI(I),I=1,LSS)
C3060       RETURN
C3070       END
C3075C ***************************************************************
C3080       SUBROUTINE INVBAN(LS,NMODE,NBAND,NZ,A,Z,WORK)
C3081C ***************************************************************
C3082C
C3083C THIS ROUTINE IS A MODIFICATION OF A STANDARD INVERSION ROUTINE
C3084C   (FOR SMALL MATRICES) USING LU DECOMPOSITION WITHOUT PIVOTING.
C3085C   ALL REFERENCES TO ELEMENTS WERE CHANGED TO SUBMATRIX REFERENCES,
C3086C   AS DESCRIBED IN THE BOTINV PROGRAM DESCIPTION.  ALL COMMENTS IN
C3087C   THIS PROGRAM REFERENCE THE ORIGINAL PROGRAM.
C3088C
C3090C THIS ROUTINE WILL INVERT A BANDED MATRIX. NBAND-1 IS THE NUMBER OF
C3100C BANDS ABOVE AND BELOW THE MAIN DIAGONAL.   ONLY THE BANDED PORTION
C3110C OF MATRIX A IS STORED BY COLUMNS.
C3120       COMPLEX A(1),Z(1),WORK(1)
```

```
      INTEGER NZ(1)
C NZ(I) = NZ(I-1) + THE NUMBER OF ZEROES BELOW THE BAND IN COLUMN I-1
C + THE NUMBER OF ZEROES ABOVE THE BAND IN COLUMN I, WHERE NZ(1)=0.
      COMMON NM,JK(4),LR(1)
      K1=NBAND
      K2=N-NBAND+2
C
C NOTE, THE LOCATION OF ELEMENT A(I,J) IN THE BANDED PORTION OF MATRIX
C A IS GIVEN BY  N * (J-1) + I - NZ(J).
C
C PERFORM THE LU DECOMPOSITION OF A, WHERE U HAS UNIT DIAGONAL ELEMENTS.
      N=2*NMODE-1
      LSS=LS*LS
      DO 100 J=1,N
      J1=J-1
      JP=J+1
      IL=(N*J1+J1-NZ(J))*LSS+1
C CALCULATE L(I,J).
      DO 20 I=J,N
      IF(IABS(I-J).GE.NBAND) GO TO 20
      IF(J.EQ.1) GO TO 19
      DO 15 K=1,J1
      IF(IABS(I-K).GE.NBAND .OR. IABS(K-J).GE.NBAND) GO TO 15
      KL=(N*(K-1)+I-NZ(K)-1)*LSS+1
      KU=(N*(J-1)+K-NZ(J)-1)*LSS+1
      CALL MULTS(LS,A(IL),A(KL),A(KU))
15    CONTINUE
19    CONTINUE
      IF(I.EQ.J) CALL LINEQ(LS,A(IL),LR)
20    IL=IL+LSS
C CALCULATE U(J,I).
      IF(J.EQ.N) GO TO 100
      IL=(N*(J-1)+J-NZ(J)-1)*LSS+1
      DO 30 I=JP,N
      IF(IABS(I-J).GE.NBAND) GO TO 30
      IU=(N*(I-1)+J-NZ(I)-1)*LSS+1
      IF(J.EQ.1) GO TO 29
      DO 25 K=1,J1
      IF(IABS(I-K).GE.NBAND .OR. IABS(K-J).GE.NBAND) GO TO 25
      KL=(N*(K-1)+I-NZ(K)-1)*LSS+1
      KU=(N*(J-1)+K-NZ(J)-1)*LSS+1
      CALL MULTS(LS,A(IU),A(KL),A(KU))
25    CONTINUE
29    CALL MULT(LS,WORK,A(IL),A(IU))
30    CONTINUE
100   CONTINUE
C FIND THE INVERSE OF A BY COLUMNS.  SOLVE LZ=B, WHERE B IS A COLUMN
C OF THE IDENTITY MATRIX.  THEN SOLVE UX=Z.  X WILL THEN BE THE
C CORRESPONDING COLUMN OF THE INVERSE MATRIX.
      M2=-NMODE
      DO 200 J=1,N
      M2=M2+1
```

```
C3640C CALCULATE Z FOR COLUMN J OF THE IDENTITY MATRIX.
C3650       IL=(N*(J-1)+J-NZ(J)-1)*LSS+1
C3660       CALL REPLACE(LS,Z((J-1)*LSS+1),A(IL))
C3670       IF(J.EQ.N) GO TO 120
C3680       JP=J+1
C3690       DO 115 I=JP,N
C3700       CALL ZERO(LS,Z((I-1)*LSS+1))
C3710       IL=(N*(I-1)+I-NZ(I)-1)*LSS+1
C3720       I1=I-1
C3730       DO 110 K=J,I1
C3740       IF(IABS(K-I).GE.NBAND) GO TO 110
C3750       KL=(N*(K-1)+I-NZ(K)-1)*LSS+1
C3760       CALL MULTS(LS,Z((I-1)*LSS+1),A(KL),Z((K-1)*LSS+1))
C3770 110   CONTINUE
C3780 115   CALL MULT(LS,WORK,A(IL),Z((I-1)*LSS+1))
C3790 120   CONTINUE
C3800       L=N
C3810       DO 150 I=2,N
C3820       L=L-1
C3830       LP=L+1
C3840       IF(L.LT.J) CALL ZERO(LS,Z((L-1)*LSS+1))
C3850       DO 140 K=LP,N
C3860       IF(IABS(K-L).GE.NBAND) GO TO 140
C3870       KU=(N*(K-1)+L-NZ(K)-1)*LSS+1
C3880       CALL MULTS(LS,Z((L-1)*LSS+1),A(KU),Z((K-1)*LSS+1))
C3890 140   CONTINUE
C3900 150   CONTINUE
C3910       M1=-NMODE
C3920       DO 160 L=1,N
C3930       M1=M1+1
C3940 160   CALL LIST(M1,M2,Z((L-1)*LSS+1))
C3950 200   CONTINUE
C3960       RETURN
C3970       END
C3975C ************************************************************
C3980       SUBROUTINE MULTS(N,A,B,C)
C3985C ************************************************************
C3990C A = A - B * C
C4000       COMPLEX S,A(1),B(1),C(1)
C4010       IS=0
C4020       DO 100 J=1,N
C4030       DO 100 I=1,N
C4040       IS=IS+1
C4050       S=0.0
C4060       IR=I
C4070       IC=(J-1)*N+1
C4080       DO 50 K=1,N
C4090       S=S+B(IR)*C(IC)
C4100       IR=IR+N
C4110 50    IC=IC+1
C4120       A(IS)=A(IS)-S
C4130 100   CONTINUE
```

```
04140       RETURN
04150       END
04155C ***************************************************************
04160       SUBROUTINE MULT(N,A,B,C)
04165C ***************************************************************
04170C C = B * C
04180       COMPLEX S,A(1),B(1),C(1)
04190       IS=0
04200       DO 100 J=1,N
04210       DO 80 I=1,N
04220       S=0.0
04230       IR=I
04240       IC=(J-1)*N+1
04250       DO 50 K=1,N
04260       S=S+B(IR)*C(IC)
04270       IR=IR+N
04280 50    IC=IC+1
04290       A(I)=S
04300 80    CONTINUE
04310       DO 90 I=1,N
04320       IS=IS+1
04330 90    C(IS)=A(I)
04340 100   CONTINUE
04350       RETURN
04360       END
04365C ***************************************************************
04370       SUBROUTINE REPLACE(N,A,B)
04375C ***************************************************************
04380C A = B
04390       COMPLEX A(1),B(1)
04400       N2=N*N
04410       DO 100 I=1,N2
04420 100   A(I)=B(I)
04430       RETURN
04440       END
04445C ***************************************************************
04450       SUBROUTINE ZERO(N,A)
04455C ***************************************************************
04460C A = 0
04470       COMPLEX A(1)
04480       N2=N*N
04490       DO 100 I=1,N2
04500 100   A(I)=C.C
04510       RETURN
04520       END
04530/EOR
04540   4 14
04542    .0000    .0000    .0000    .0000    .0000    .0000    .0000    .0000    .0000    .0000
04544    .0000    .0000    .0000    .0000    .0000    .0000
04546    .0000    .0000    .0000    .0000    .0000    .0000    .0000    .0000    .0000    .0000
04548    .0000    .0000    .0000    .0000    .0000    .0000
04550/EOR
```


BOTINWA

```
CC1CCRDTI,P3C,T2C,CM24CCCC,STCX3.
CC110ACCOUNT,M0326S,3DT2A.
CC120BANNERS(OUTPUT) *J. OUTNAM*DEPT 220*RLD 111-4**
CC130ATTACH(INFIL1=YSSX)
CC131ATTACH(INFIL2=ZSWX)
CC132ATTACH(INFIL4=ZWWX)
CC133ATTACH(INFIL3=ZCWX)
CC134GOTO,DEFINE.
CC135EXIT.
CC136DEFINE(OUTFIL=YSSWX)
CC140FTN(R=0,OPT=2)
CC170LDSET(PRESET=INDEF)
CC180LGO.
CC220/EOR
CC225       PROGRAM BDTINVA(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
CC226      1 INFIL1,TAPE1=INFIL1,INFIL2,TAPE2=INFIL2,
CC228      2 INFIL4,TAPE4=INFIL4,OUTFIL,TAPE7=OUTFIL,
CC23C      3 INFIL3,TAPE3=INFIL3)
CC232C
CC233C BDTINVA INVERTS THE BDT Z MATRIX WITH WIRE AND/OR CAP ADDITIONS.
CC234C
CC235C UNIT 5 IS THE CARD READER.
CC236C UNIT 6 IS THE LINE PRINTER.
CC238C UNIT 7 IS A DISK FILE FOR OUTPUT OF THE Y MATRIX.
CC239C UNIT 1 IS A DISK FILE CONTAINING THE PI MATRIX.
CC24CC UNIT 2 IS A DISK FILE CONTAINING THE Q MATRIX.
CC241C UNIT 3 IS A DISK FILE CONTAINING PART OF THE Q MATRIX.
CC243C UNIT 4 IS A DISK FILE CONTAINING THE S MATRIX.
CC244C
CC245       COMPLEX PI(12544),Q(5376),R(5376),S(2304),YI(2304)
CC250       COMPLEX W1(115),W2(48)
CC252       COMPLEX LOAD(48)
CC254       DIMENSION WGHT(48)
CC26C       DIMENSION LAD(1)
CC27C       COMMON /ORDER/ MORD(2)
CC285       NFOUND=C
CC290       READ(1) NMODE,NP,NC,NPR,NE,NW,NPW,NJ
CC292       READ(1) MORD
CC294       WRITE(6,2) 'OLD',NMODE,NP,NC,NPR,NE,NW,NPW,NJ,MORD
CC296  2    FORMAT(1X,A4,'Y MATRIX PARAMETERS:',/,
CC298      1 18X,'NMODE   NP     NC    NPR     NE     NW    NPW     NJ',/,
CC3CC      2 15X,8I6,//,
CC3C2      3 7X,'MODE NO. OF ADDITIONS:',5X,'CAPS',5X,'WIRES',/,
CC3C4      4 28X,2I9)
CC3C6       READ(5,1) NC1,NPR1,NE1,NW1,NPW1,NJ1
CC3C8  1    FORMAT(6I3)
CC31C       IF(NE1.NE.C) NE1=NC1
CC312       WRITE(6,3) NC1,NPR1,NE1,NW1,NPW1,NJ1
CC314  3    FORMAT(//,' NEW ADDITIONS:',/,
CC316      1 18X,'NC1   NPR1    NE1     NW1    NPW1     NJ1',/,
CC318      2 14X,6I6)
CC320       KMODE=2*NMODE-1
```

```
CC322       LS=NP-3
CC324       NM=LS/2
CC326       LC=NC*NM*(NPR-3)/2
CC327       LE=NE*NM
CC328       LW=(NPW-3*NW)/2
CC330       IERA=C
CC332       IERS=0
CC334       IF(NC1.EQ.0) GO TO 20
CC44CC CAPS ADDED.
CC45C       WRITE(6,8)
CC452  8    FORMAT(//,1X,'CAPS ADDED TO MATRIX',//)
CC454       NC=NC1
CC456       NPR=NPR1
CC458       NE=NE1
CC46C       LC=NC*NM*(NPR-3)/2
CC461       LE=NE*NM
CC462       MORD(1)=NMODE
CC463       IF(MORD(2).NE.0) MORD(1)=NMODE+1
CC464       NAD=C
CC465       IF(MORD(2).NE.0) NAD=1
CC466       LAD(1)=LW+NJ
CC468       LAD1=2*LC+NE*NM
CC47C       IF(NW1.NE.0) IERA=1
CC471       READ(2) NP1,NC1,NPR1,NE1
CC472       IF(NP.NE.NP1) IERS=1
CC473       IF(NC.NE.NC1) IERS=2
CC474       IF(NPR.NE.NPR1) IERS=3
CC475       IF(NE.NE.NE1) IERS=4
CC476       READ(4) NC1,NPR1,NE1
CC477       IF(NC.NE.NC1) IERS=5
CC478       IF(NPR.NE.NPR1) IERS=6
CC479       IF(NE.NE.NE1) IERS=7
CC48C       IF(MORD(2).EQ.0) GO TO 15
CC481       READ(3) NC1,NPR1,NE1,NW1,NPW1,NJ1
CC482       IF(NC.NE.NC1) IERS=8
CC483       IF(NPR.NE.NPR1) IERS=9
CC484       IF(NE.NE.NE1) IERS=10
CC485       IF(NW.NE.NW1) IERS=11
CC486       IF(NPW.NE.NPW1) IERS=12
CC487       IF(NJ.NE.NJ1) IERS=13
CC488 15    CONTINUE
CC490C READ SURFACE IMPEDANCE LOADING.
CC492       READ(5,11) (W2(I),I=1,NM)
CC494  11   FORMAT(10F8.4)
CC496       NM1=NM+1
CC498       READ(5,11) (W2(I),I=NM1,LS)
CC5CC       WRITE(6,12)
CC5C2  12   FORMAT(/,'  I AND Z SURFACE IMPEDANCE LOADING (COMPLEX)')
CC5C4       WRITE(6,59) (W2(I),I=1,NM)
CC5C6  59   FORMAT(//,(1X,10F8.4))
CC5C8       WRITE(6,59) (W2(I),I=NM1,LS)
CC51C       LR=(NPR-3)/2
```

```
C0528         J=0
C0529         DO 100 JC=1,NC
C0530         DO 100 JP=1,NM
C0532         DO 100 JR=1,LR
C0534         J=J+1
C0536         LOAD(J)=W2(JP)
C0537  100    LOAD(J+LC)=W2(JP+NM)
C0538         IF(NE.EQ.0) GO TO 120
C0539         J=2*LC
C0540         DO 110 JC=1,NC
C0542         DO 110 JP=1,NM
C0544         J=J+1
C0546  110    LOAD(J)=W2(JP+NM)
C0548  120    CONTINUE
C0550         I1=NP*LR
C0560         READ(4) (WGHT(I),I=1,I1)
C0570         J=0
C0580         DO 130 JC=1,NC
C0590         I=0
C0600         DO 130 JP=1,NM
C0610         DO 130 JR=1,LR
C0620         J=J+1
C0630         I=I+1
C0640  130    LOAD(J)=LOAD(J)*WGHT(I)
C0650         READ(4) (WGHT(I),I=1,I1)
C0660         DO 140 JC=1,NC
C0670         I=0
C0680         DO 140 JP=1,NM
C0690         DO 140 JR=1,LR
C0700         J=J+1
C0710         I=I+1
C0720  140    LOAD(J)=LOAD(J)*WGHT(I)
C0730         IF(NE.EQ.0) GO TO 160
C0740         READ(4) (WGHT(I),I=1,NM)
C0750         DO 150 JC=1,NC
C0760         I=0
C0770         DO 150 JP=1,NM
C0780         J=J+1
C0790         I=I+1
C0800  150    LOAD(J)=LOAD(J)*WGHT(I)
C0810  160    CONTINUE
C0812         GO TO 50
C0814C WIRES ADDED.
C0820  20     WRITE(6,9)
C0830  9      FORMAT(//,1X,'WIRES ADDED TO MATRIX',//)
C0840         NW=NW1
C0850         NPW=NPW1
C0860         NJ=NJ1
C0870         LW=(NPW-3*NW)/2
C0880         MORD(2)=NMODE
C0882         IF(MORD(1).NE.0) MORD(2)=NMODE+1
C0884         NAD=0
```

```
      IF(MCRD(1).NE.0) NAD=1
      LAD(1)=2*LC+NE*NM
      LAD1=LW+NJ
      IF(NC1.NE.0) IERA=1
      READ(2) NP1,NW1,NPW1,NJ1
      IF(NP.NE.NP1) IERS=1
      IF(NW.NE.NW1) IERS=2
      IF(NPW.NE.NPW1) IERS=3
      IF(NJ.NE.NJ1) IERS=4
      READ(4) NW1,NPW1,NJ1
      IF(NW.NE.NW1) IERS=5
      IF(NPW.NE.NPW1) IERS=6
      IF(NJ.NE.NJ1) IERS=7
      IF(MCRD(1).EQ.0) GO TO 50
      READ(3) NC1,NPR1,NE1,NW1,NPW1,NJ1
      IF(NC.NE.NC1) IERS=8
      IF(NPR.NE.NPR1) IERS=9
      IF(NE.NE.NE1) IERS=10
      IF(NW.NE.NW1) IERS=11
      IF(NPW.NE.NPW1) IERS=12
      IF(NJ.NE.NJ1) IERS=13
   50 CONTINUE
      WRITE(6,2) 'NEW',NMODE,NP,NC,NPR,NE,NW,NPW,NJ,MORD
      IF(IERA.NE.0) GO TO 900
      IF(IERS.NE.0) GO TO 900
      NEXP=(KMODE+NAD)**2+KMODE+NAD+1
      LL=KMODE*LS
      IF(NAD.EQ.0) GO TO 250
      DO 200 I=1,NAD
  200 LL=LL+LAD(I)
  250 CONTINUE
C DETERMINE MINIMUM DIMENSIONS.
      K1=LL*LL
      K2=LL*LAD1
      K3=K2
      K4=LAD1*LAD1
      K5=MAX0(LS*LS,LS*(LW+NJ),LS*(2*LC+LE),(LW+NJ)**2,
     1  (2*LC+LE)**2,(LW+NJ)*(2*LC+LE))
      K6=MAX0(LL,LAD1)
      K7=LAD1
      WRITE(6,5) K1,K2,K3,K4,K5,K6,K7
    5 FORMAT(//' MINIMUM PROGRAM DIMENSIONS ARE AS FOLLOWS:',/,
     1 '      PI       Q        P        S       YI       WI',
     2 '       W2,LOADEWGHT',/,7I8)
      CALL GETPQRS(NMODE,NAD,LS,LAD,LAD1,W2,PI,Q,R,S,YI,NFOUND)
      IF(NFOUND.NE.NEXP) GO TO 490
      IF(MCRD(1).LT.MORD(2)) GO TO 112
      WRITE(6,113)
  113 FORMAT(//,'  CAP IMPEDANCE ADDED')
C ADD CAP IMPEDANCE TO DIAGONAL.
      I=1
      DO 111 J=1,LAD1
```

```
C1410         S(I)=S(I)+LOAD(J)
C1420   111   I=I+LAD1+1
C1430   112   CONTINUE
C1440C INVERT THE Z MATRIX.
C1450         CALL INVPAR(PI,Q,R,S,W1,W2,LL,LAD1)
C1460C OUTPUT Z MATRIX BY MODE NUMBERS.
C1470         WRITE(7) NMODE,NP,NC,NPR,NF,NW,NPW,NJ
C1480         WRITE(7) MODE
C1490         CALL OUTPORS(NMODE,NAD,LS,LAD,LAD1,W2,PI,Q,R,S,YI)
C1500         GO TO 500
C1510  490   WRITE(6,7)
C1520  7      FORMAT(///,41H ERROR IN THE NUMBER OF SUBMATRICES FOUND)
C1530  500    CONTINUE
C1540         WRITE(6,4) NEXP,NFOUND
C1550  4      FORMAT(///,33H NUMBER OF SUBMATRICES EXPECTED =,I5,///,
C1560     1  30H NUMBER OF SUBMATRICES FOUND =,I8)
C1570         STOP
C1580C ERROR IN DATA INPUT.
C1590  900    CONTINUE
C1600         IF(IERR.NE.0) WRITE(6,901)
C1610  901    FORMAT(///,' *** ERROR *** BOTH CARS AND WIRES CAN NOT BE',
C1620     1  ' ADDED AT THE SAME TIME')
C1630         IF(IERS.NE.0) WRITE(6,902) IERS
C1640  902    FORMAT(///,' *** ERROR *** ONE OR MORE OF THE INPUT ',
C1650     1  'PARAMETERS DO NOT AGREE WITH Z MATRIX FILES',
C1660     2  3X,'CODE =',I4)
C1670         STOP
C1680         END
C1690C ***************************************************************
C1700         SUBROUTINE LIST(M,N,ZI,LI,LJ)
C1710C ***************************************************************
C1720C PRINT Z(M,N) ON THE LINE PRINTER AND WRITE TO DISK FILE.
C1730         DIMENSION JK(4)
C1740         COMPLEX ZI(1)
C1750         NI=LI/2
C1760         NJ=LJ/2
C1770         LSS=LI*LJ
C1780         IF(M.NE.N) GO TO 200
C1790         JK(1)=1
C1800         JK(2)=NI+1
C1810         JK(3)=2*NI*NJ+1
C1820         JK(4)=JK(3)+NI
C1830         WRITE(6,2) M,N
C1840  2      FORMAT(4H1 M=,I3,4H  N=,I3,/)
C1850         DO 140 J=1,4
C1860         K1=JK(J)
C1870         WRITE(6,3) J
C1880  3      FORMAT(2H Y,I1)
C1890         DO 140 I=1,NJ
C1900         K2=K1+NI-1
C1910         WRITE(6,4) (ZI(K),K=K1,K2)
C1920  4      FORMAT(1X,10G11.4)
```

```
      K1=K1+LI
  140 CONTINUE
  200 CONTINUE
      WRITE(7) M,N
      WRITE(7) (ZI(I),I=1,LSS)
      RETURN
      END
C-----------------------------------------------------------
      SUBROUTINE INVPAR(PI,Q,R,S,W,LP,N,M)
C
C  WRITTEN 8/8/80 BY JOHN M. PUTNAM DEPT 220
C
C  INVERT THE PARTITIONED MATRIX P Q WHERE P INVERSE(PI) HAS
C                               R S
C  ALREADY BEEN COMPUTED.  N AND M ARE THE ORDER OF P AND S, RESPECTIVELY.
C  ON RETURN, THESE ARRAYS CONTAIN THE INVERSE.
C  W AND LP ARE WORK ARRAYS OF DIMENSION MAX(N,M) AND M, RESPECTIVELY.
C
      COMPLEX PI(1),Q(1),R(1),S(1),W(1)
      DIMENSION LP(1)
      CALL ASQ(R,PI,W,M,N,+1)
      CALL ABC(S,P,Q,M,N)
      CALL LINEQ(M,S,LP)
      CALL SQA(PI,Q,W,N,M,+1)
      CALL ASQ(Q,S,W,N,M,-1)
      CALL ABC(PI,Q,P,N,M)
      CALL SQA(S,P,W,M,N,-1)
      RETURN
      END
C-----------------------------------------------------------
      SUBROUTINE SQA(SQ,A,W,N,M,I)
C  FORM THE PRODUCT I*SQ*A, WHERE SQ IS A SQUARE MATRIX OF ORDER N.
C  THE RESULT IS RETURNED IN A.
      COMPLEX SQ(1),A(1),W(1),WW
      DO 400 JM=1,M
      DO 200 IN=1,N
      IA=(JM-1)*N+1
      ISQ=IN
      WW=0.
      DO 100 JN=1,N
      WW=WW+SQ(ISQ)*A(IA)
      ISQ=ISQ+N
      IA=IA+1
  100 CONTINUE
      W(IN)=I*WW
  200 CONTINUE
      IA=(JM-1)*N+1
      DO 300 IN=1,N
      A(IA)=W(IN)
      IA=IA+1
  300 CONTINUE
  400 CONTINUE
```

```
      RETURN
      END
C-----------------------------------------------------------
      SUBROUTINE ASQ(A,SQ,W,M,N,I)
C FORM THE PRODUCT T*A*SQ, WHERE SQ IS A SQUARE MATRIX OF ORDER N.
C THE RESULT IS RETURNED IN A.
      COMPLEX A(1),SQ(1),W(1),WW
      DO 400 IM=1,M
      DO 200 JN=1,N
      IA=IM
      ISQ=(JN-1)*N+1
      WW=0.
      DO 100 IN=1,N
      WW=WW+SQ(ISQ)*A(IA)
      ISQ=ISQ+1
      IA=IA+M
  100 CONTINUE
      W(JN)=I*WW
  200 CONTINUE
      IA=IM
      DO 300 JN=1,N
      A(IA)=W(JN)
      IA=IA+M
  300 CONTINUE
  400 CONTINUE
      RETURN
      END
C-----------------------------------------------------------
      SUBROUTINE ABC(A,B,C,N,M)
C COMPUTE A = A - B * C, WHERE A IS A SQUARE MATRIX OF ORDER N.
      COMPLEX A(1),B(1),C(1),WW
      IA=0
      DO 400 JN=1,N
      DO 300 IN=1,N
      IB=IN
      IC=(JN-1)*M+1
      WW=0.
      DO 200 KM=1,M
      WW=WW+B(IB)*C(IC)
      IB=IB+N
      IC=IC+1
  200 CONTINUE
      IA=IA+1
      A(IA)=A(IA)-WW
  300 CONTINUE
  400 CONTINUE
      RETURN
      END
C-----------------------------------------------------------
      SUBROUTINE CINV3(LL,C,LA)
C COMPLEX MATRIX INVERSION ROUTINE.
      COMPLEX C(1),SIGN,SIG,ST,S
```

```
C2970       DIMENSION LR(1)
C2980       DO 20 I=1,LL
C2990       LR(I)=I
C3000    20 CONTINUE
C3010       M1=0
C3020       DO 18 M=1,LL
C3030       K=M
C3040       K2=M1+K
C3050       S1=CABS(C(K2))
C3060       DO 2 I=M,LL
C3070       K1=M1+I
C3080       S2=CABS(C(K1))
C3090       IF(S2-S1) 2,2,6
C3100     6 K=I
C3110       S1=S2
C3120     2 CONTINUE
C3130       LS=LR(M)
C3140       LR(M)=LR(K)
C3150       LR(K)=LS
C3160       K2=M1+K
C3170       STOR=C(K2)
C3180       J1=0
C3190       DO 7 J=1,LL
C3200       K1=J1+K
C3210       K2=J1+M
C3220       STO=C(K1)
C3230       C(K1)=C(K2)
C3240       C(K2)=STO/STOR
C3250       J1=J1+LL
C3260     7 CONTINUE
C3270       K1=M1+M
C3280       C(K1)=1./STOR
C3290       DO 11 I=1,LL
C3300       IF(I-M) 12,11,12
C3310    12 K1=M1+I
C3320       ST=C(K1)
C3330       C(K1)=0.
C3340       J1=0
C3350       DO 10 J=1,LL
C3360       K1=J1+I
C3370       K2=J1+M
C3380       C(K1)=C(K1)-C(K2)*ST
C3390       J1=J1+LL
C3400    10 CONTINUE
C3410    11 CONTINUE
C3420       M1=M1+LL
C3430    18 CONTINUE
C3440       J1=0
C3450       DO 9 J=1,LL
C3460       IF(J-LR(J)) 14,8,14
C3470    14 LRJ=LR(J)
C3480       J2=(LRJ-1)*LL
```

```
C3490      21 DO 13 I=1,LL
C3500         K2=J2+I
C3510         K1=J1+I
C3520         S=C(K2)
C3530         C(K2)=C(K1)
C3540         C(K1)=S
C3550      13 CONTINUE
C3560         LR(J)=LR(LRJ)
C3570         LR(LRJ)=LRJ
C3580         IF(J-LR(J)) 14,8,14
C3590       8 J1=J1+LL
C3600       9 CONTINUE
C3610         RETURN
C3620         END
C3630C ------------------------------------------------
C3640         SUBROUTINE GETPQRS(NMODE,NAD,LS,LAD,LAD1,LWORK,P,Q,R,S,YI,NF)
C3650         DIMENSION LAD(1),LWORK(1)
C3660         COMPLEX P(1),Q(1),R(1),S(1),YI(1)
C3670         COMMON /ORDER/ MORD(2)
C3680         KMODE=2*NMODE-1
C3690         LL=KMODE*LS
C3700         LSS=LS*LS
C3710         IF(NAD.EQ.0) GO TO 40
C3720         DO 20 I=1,NAD
C3730         LWORK(I)=LL
C3740      20 LL=LL+LAD(I)
C3750      40 CONTINUE
C3760C RETRIEVE MATRIX P BY SUBMATRICES.
C3770     100 READ(1) M,N
C3780         IF(EOF(1)) 200,110
C3790     110 CONTINUE
C3800         NF=NF+1
C3810         IF(M.GE.NMODE) GO TO 130
C3820         IF(N.GE.NMODE) GO TO 150
C3830         READ(1) (YI(I),I=1,LSS)
C3840         II=0
C3850         K=(N+NMODE-1)*LL*LS+(M+NMODE-1)*LS
C3860         DO 120 J=1,LS
C3870         DO 115 I=1,LS
C3880         II=II+1
C3890         K=K+1
C3900         P(K)=YI(II)
C3910     115 CONTINUE
C3920         K=K+LL-LS
C3930     120 CONTINUE
C3940         GO TO 100
C3950     130 CONTINUE
C3960         IF(N.GE.NMODE) GO TO 170
C3970         LI=LAD(M-NMODE+1)
C3980         LIJ=LI*LS
C3990         READ(1) (YI(I),I=1,LIJ)
C4000         II=0
```

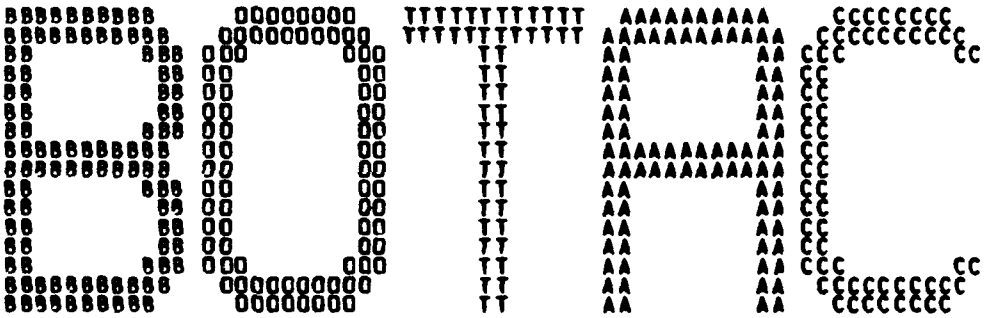
```
C4v1C         K=(N+NMODE-1)*LL*LS+LWORK(M-NMODE+1)
C4020         DO 140 J=1,LS
C4030         DO 135 I=1,LI
C4040         II=II+1
C4050         K=K+1
C4060         P(K)=YI(II)
C4070  135    CONTINUE
C4080         K=K+LL-LI
C4090  140    CONTINUE
C4100         GO TO 100
C4110  150    CONTINUE
C4120         LJ=LAD(N-NMODE+1)
C4130         LIJ=LS*LJ
C4140         READ(1) (YI(I),I=1,LIJ)
C4150         II=0
C4160         K=LWORK(N-NMODE+1)*LL+(M+NMODE-1)*LS
C4170         DO 160 J=1,LJ
C4180         DO 155 I=1,LS
C4190         II=II+1
C4200         K=K+1
C4210         P(K)=YI(II)
C4220  155    CONTINUE
C4230         K=K+LL-LS
C4240  160    CONTINUE
C4250         GO TO 100
C4260  170    CONTINUE
C4270         LI=LAD(M-NMODE+1)
C4280         LJ=LAD(N-NMODE+1)
C4290         LIJ=LI*LJ
C4300         READ(1) (YI(I),I=1,LIJ)
C4310         II=0
C4320         K=LWORK(N-NMODE+1)*LL+LWORK(M-NMODE+1)
C4330         DO 180 J=1,LJ
C4340         DO 175 I=1,LI
C4350         II=II+1
C4360         K=K+1
C4370         P(K)=YI(II)
C4380  175    CONTINUE
C4390         K=K+LL-LI
C4400  180    CONTINUE
C4410         GO TO 100
C4420C RETRIEVE R & Q RY SUBMATRICES, USING THE SYMMETRY RELATION
C4430C BETWEEN R AND Q.  R(N) = Q(-N) TRANSPOSE.
C4440  200    CONTINUE
C4450  300    READ(2) M
C4460         IF(ECOF(2)) 400,310
C4470  310    CONTINUE
C4490         LIJ=LS*LAD1
C4500         READ(2) (YI(I),I=1,LIJ)
C4505         IF(IABS(M).GE.NMODE) GO TO 300
C4510         NF=NF+1
C4520         II=0
```

```
C4530       K=(M+NMODE-1)*LS
C4540       DO 320 J=1,LAD1
C4550       KT=(-M+NMODE-1)*LAD1*LS+J
C4560       DO 315 I=1,LS
C4570       II=II+1
C4580       K=K+1
C4590       Q(K)=YI(II)
C4600       R(KT)=YI(II)
C4610       KT=KT+LAD1
C4620 315   CONTINUE
C4630       K=K+LL-LS
C4640 320   CONTINUE
C4650       GO TO 300
C4670C RETRIEVE ADDITIONAL PART OF MATRIX Q.
C4680  400  CONTINUE
C4690       IF(LAD(1).EQ.0) GO TO 500
C4700       LIJ=LAD(1)*LAD1
C4710       READ(3) (YI(I),I=1,LIJ)
C4720       NF=NF+1
C4730       DO 420 J=1,LAD1
C4740       LADI=LAD(1)
C4750       DO 415 I=1,LADI
C4760       K=(J-1)*(KMODE*LS+LADI)+KMODE*LS+I
C4770       KT=KMODE*LS*LAD1+(I-1)*LAD1+J
C4780       II=(J-1)*LADI+I
C4790       IF(MORD(1).NE.NMODE) II=(I-1)*LAD1+J
C4800       Q(K)=YI(II)
C4810       R(KT)=YI(II)
C4820 415   CONTINUE
C4830 420   CONTINUE
C4840C RETRIEVE MATRIX S.
C4850  500  LL2=LAD1*LAD1
C4860       READ(4) (S(I),I=1,LL2)
C4870       NF=NF+1
C4880       RETURN
C4890       END
C4900C ------------------------------------------------------
C4910       SUBROUTINE PUTPQRS(NMODE,NAD,LS,LAD,LAD1,LWORK,P,Q,R,S,YI)
C4920       DIMENSION LAD(1),LWORK(1)
C4930       COMPLEX P(1),Q(1),R(1),S(1),YI(1)
C4940       KMODE=2*NMODE-1
C4950       LL=KMODE*LS
C4960       IF(NAD.EQ.0) GO TO 40
C4970       DO 20 I=1,NAD
C4980       LWORK(I)=LL
C4990  20   LL=LL+LAD(I)
C5000  40   CONTINUE
C5010C STORE MATRIX P BY SUBMATRICES.
C5020       DO 100 NN=1,KMODE
C5030       N=-NMODE+NN
C5040       DO 100 MM=1,KMODE
C5050       M=-NMODE+MM
```

```
C5066C          II=C
C5070           K=(N+NMODE-1)*LL*LS+(M+NMODE-1)*LS
C5080           DO 8C J=1,LS
C5090           DO 6C I=1,LS
C5100           II=II+1
C5110           K=K+1
C5120           YI(II)=P(K)
C5130  6C       CONTINUE
C5140           K=K+LL-LS
C5150  8C       CONTINUE
C5160           CALL LIST(M,N,YI,LS,LS)
C5170  1CC      CONTINUE
C5180           IF(NAD.EQ.C) GO TO 420
C5190           DO 2CC NN=1,KMODE
C5200           N=-NMODE+NN
C5210           DO 2CC MM=1,NAD
C5220           M=NMODE-1+MM
C5230           II=C
C5240           K=(N+NMODE-1)*LL*LS+LWORK(MM)
C5250           LI=LAD(MM)
C5260           DO 18C J=1,LS
C5270           DO 16C I=1,LI
C5280           II=II+1
C5290           K=K+1
C5300           YI(II)=P(K)
C5310  16C      CONTINUE
C5320           K=K+LL-LI
C5330  18C      CONTINUE
C5340           CALL LISTA(M,N,YI,LI,LS)
C5350  2CC      CONTINUE
C5360           DO 36C NN=1,NAD
C5370           N=NMODE-1+NN
C5380           DO 3CC MM=1,KMODE
C5390           M=-NMODE+MM
C5400           II=C
C5410           K=LWORK(NN)*LL+(M+NMODE-1)*LS
C5420           LJ=LAD(NN)
C5430           DO 28C J=1,LJ
C5440           DO 26C I=1,LS
C5450           II=II+1
C5460           K=K+1
C5470           YI(II)=P(K)
C5480  26C      CONTINUE
C5490           K=K+LL-LS
C5500  28C      CONTINUE
C5510           CALL LISTA(M,N,YI,LS,LJ)
C5520  3CC      CONTINUE
C5530           DO 4CC NN=1,NAD
C5540           N=NMODE-1+NN
C5550           DO 4CC MM=1,NAD
C5560           M=NMODE-1+MM
C5570           II=C
```

```
C5580         K=LWORK(NN)*LL+LWORK(MM)
C5590         LI=LAD(MM)
C5600         LJ=LAD(NN)
C5610         DO 380 J=1,LJ
C5620         DO 360 I=1,LI
C5630         II=II+1
C5640         K=K+1
C5650         YI(II)=P(K)
C5660  360    CONTINUE
C5670         K=K+LL-LI
C5680  380    CONTINUE
C5690         CALL LISTA(M,N,YI,LI,LJ)
C5700  400    CONTINUE
C5710  420    CONTINUE
C5720C STORE MATRIX R BY SUBMATRICES.
C5730         M=NMODE+NAD
C5740         DO 500 NN=1,KMODE
C5750         N=-NMODE+NN
C5760         K=(N+NMODE-1)*LAD1*LS
C5770         CALL LISTA(M,N,P(K+1),LAD1,LS)
C5780  500    CONTINUE
C5790         IF(NAD.EQ.0) GO TO 610
C5800         DO 600 NN=1,NAD
C5810         N=NMODE-1+NN
C5820         K=LWORK(NN)*LAD1
C5830         CALL LISTA(M,N,P(K+1),LAD1,LAD(NN))
C5840  600    CONTINUE
C5850  610    CONTINUE
C5860C STORE MATRIX Q BY SUBMATICES.
C5870         N=NMODE+NAD
C5880         DO 700 MM=1,KMODE
C5890         M=-NMODE+MM
C5900         II=0
C5910         K=(M+NMODE-1)*LS
C5920         LJ=LAD1
C5930         DO 680 J=1,LJ
C5940         DO 660 I=1,LS
C5950         II=II+1
C5960         K=K+1
C5970         YI(II)=Q(K)
C5980  660    CONTINUE
C5990         K=K+LL-LS
C6000  680    CONTINUE
C6010         CALL LISTA(M,N,YI,LS,LJ)
C6020  700    CONTINUE
C6030         IF(NAD.EQ.0) GO TO 810
C6040         DO 800 MM=1,NAD
C6050         M=NMODE-1+MM
C6060         II=0
C6070         K=LWORK(MM)
C6080         LI=LAD(MM)
C6090         LJ=LAD1
```

```
C6100       DO 780 J=1,LJ
C6110       DO 760 I=1,LI
C6120       II=II+1
C6130       K=K+1
C6140       YI(II)=C(K)
C6150 760   CONTINUE
C6160       K=K+LL-LI
C6170 780   CONTINUE
C6180       CALL LISTA(M,N,YI,LI,LJ)
C6190 800   CONTINUE
C6200 810   CONTINUE
C6210C STORE MATRIX S.
C6220       M=NMODE+NAD
C6230       N=NMODE+NAD
C6240       CALL LISTA(M,N,S,LAD1,LAD1)
C6250       RETURN
C6260       END
C6270C ********************************************************
C6280       SUBROUTINE LISTA(M,N,ZI,LI,LJ)
C6290C ********************************************************
C6300C PRINT ZI(M,N) ON THE LINE PRINTER AND WRITE TO DISK FILE.
C6310       COMPLEX ZI(1)
C6320       LSS=LI*LJ
C6330       IF(M.NE.N) GO TO 200
C6340       WRITE(6,2) M,N
C6350 2     FORMAT(4H1 M=,I3,4H  N=,I3,/)
C6360       K1=1
C6370       WRITE(6,3)
C6380 3     FORMAT(2H Y)
C6390       DO 140 I=1,LJ
C6400       K2=K1+LI-1
C6410       WRITE(6,4) (ZI(K),K=K1,K2)
C6420 4     FORMAT(1X,10G11.4)
C6430       K1=K1+LI
C6440 140   CONTINUE
C6450 200   CONTINUE
C6460       WRITE(7) M,N
C6470       WRITE(7) (ZI(I),I=1,LSS)
C6480       RETURN
C6490       END
C6500/EOR
C6510  C     C  1  9  1
C6520   .0000   .0000   .0000   .0000   .0000   .0000   .0000   .0000   .0000   .0000
C6530   .0000   .0000   .0000   .0000   .0000   .0000
C6540   .0000   .0000   .0000   .0000   .0000   .0000   .0000   .0000   .0000   .0000
C6550   .0000   .0000   .0000   .0000   .0000   .0000
C6560/EOR
```



```
CC100CBOTA,P30,T100,CM150000,STCX3.
CC110ACCOUNT,M0326S,BOT2A.
CC120BANNERS(OUTPUT) *J. OUTNAM*DEPT 220*BLD 110-4**
CC130ATTACH(INFIL=YSSX)
CC140FTN(R=0,OPT=1)
CC164LDSET(PRESET=INDEF)
CC170LGO.
CC190/EOR
CC200       PROGRAM BOTAC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
CC210      1 INFIL,TAPE1=INFIL)
CC220C
CC230C BOTAC IS THE BOT ANTENNA COUPLING CODE.
CC240C
CC250C UNIT 5 IS THE CARD READER.
CC260C UNIT 6 IS THE LINE PRINTER.
CC270C UNIT 1 IS A DISK FILE CONTAINING THE Y MATRIX.
CC280C
CC285       COMPLEX U,A3,A4
CC295       COMPLEX TAB(400),ZAB(400),WAB(100)
CC300       COMPLEX TCOEFF,ZCOEFF
CC305       COMPLEX YTT,YZZ
CC320       COMPLEX Y(2304)
CC326       COMMON /PLOT1/ NPLOT,XPLOT(200),YPLOT(200),ZPLOT(200),ISYM(200)
CC327       COMMON /WAVE/ BK
CC330       COMMON /BOT1/ NMODE,NPT,NBAND
CC340       COMMON /BOT2/ NP,RL,YB(83),XB(83),Y81(82),X81(82)
CC350       COMMON /BOT3/ D4(82),SV(82),CV(82)
CC370       COMMON /BOT5/ T(160),TP(160),TZ(160)
CC380       COMMON /BOT6/ IEDGE,IUNIF
CC381       COMMON /SLOT1/ NSA,IS(20),Z0(20),Z1(20)
CC382       COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
CC384      1 XW1(100),YW1(100),ZW1(100)
CC385       COMMON /WIRE4/ LW,IW(196),IPW(196),INDIW(49)
CC386       COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
CC42C       DIMENSION MORD(2)
CC425       NPLOT=0
CC430       U=(0.,1.)
CC44C       ETA=376.707
CC45C       PI=3.14159265
CC462       READ(5,1) BK
CC463  1    FORMAT(E15.7)
CC464       WRITE(6,2) BK
CC465  2    FORMAT(9H1       BK,/,E15.7)
CC470C
CC48C       CALL BOTIN
CC490C
CC5CC       CALL CAPIN
CC510C
CC520       CALL WIREIN
CC522C
CC540       CALL SLOTIN
CC55CC
```

```
CC710       LS=NP-3
CC712       NM=LS/2
CC720       LSS=LS*LS
CC735       LWJ=LW+NJ
CC740       LSW=LS*LWJ
CC770       LWW=LWJ**2
CC810       BKL=BK*BL
CC830       KMODE=2*NMODE-1
CC880       CALL ORDER(MORD)
CC890C CALCULATE ANTENNA COUPLING COEFFICIENTS.
CC891       K=0
CC892       DO 9C J=1,NSA
CC893       DO 9C I=1,NSA
CC894       K=K+1
CC895       TAB(K)=C.
CC896       ZAB(K)=C.
CC897  9C    CONTINUE
CC9CC       NUMB=C
CC91C  5C    CONTINUE
CC920       READ(1) M,N
CC93C       IF(ECF(1)) 37,60
CC940  6C    NUMB=NUMB+1
CC950C DETERMINE SUBMATRIX TYPE.
CC96C       IF(M.LT.NMODE .AND. N.LT.NMODE) GO TO 8C
CC97C       IF(M.EQ.MORD(2) .AND. N.EQ.MORD(2)) GO TO 18C
CC98C       READ(1) Y(1)
CC99C       GO TC 5C
C1122C SLOT-SLOT ANTENNA COUPLING.
C1123  8C    CONTINUE
C1124       READ(1) (Y(I),I=1,LSS)
C1125       IF(NSA.EQ.C) GO TO 50
C1158       K=0
C1160       DO 15C J=1,NSA
C1162       IB=IS(J)
C1164       DO 14C I=1,NSA
C1166       IA=IS(I)
C1168       YTT=Y(LS*(IB-1)+IA)
C1170       YZZ=Y(LS*(NM+IB-1)+NM+IA)
C1172       K=K+1
C1176       CALL MINT(M,ZC(I),Z1(I),A3,A4)
C1178       TAB(K)=TAB(K)+YTT*A3
C1180       ZAB(K)=ZAB(K)+YZZ*A4
C1214  14C   CONTINUE
C1216  15C   CONTINUE
C1218       GO TC 5C
C1264C WIRE-WIRE ANTENNA COUPLING.
C1266  18C   CONTINUE
C1267       READ(1) (Y(I),I=1,LWW)
C1268       IF(NJ.EC.0) GO TO 50
C1279       K=C
C1280       DO 21C JA=1,NJ
C1282       DO 21C JB=1,NJ
```

```
C1283         K=K+1
C1284         WAB(K)=Y(LWJ*(LW+JB-1)+LW+JA)
C1288  210    CONTINUE
C1544         GO TO 50
01550C ALL Y SUBMATRICES HAVE BEEN READ.
01560  37     CONTINUE
C1562         WRITE(6,114) NUMQ
C1564  114    FORMAT(//,1X,I4,17H SUBMATRICES READ)
01570C PRINT THE COUPLING COEFFICIENTS.
01590         WRITE(6,101)
C1600  101    FORMAT(//,' SLOT-SLOT ANTENNA COUPLING',//,
C1610       1 '    SLOT A    SLOT B    T AND Z COUPLING COEFFS. (COMPLEX)',/)
C1640         K=0
C1650         DO 185 J=1,NSA
C1651         IB=IS(J)
C1652         TCOEFF=0.0
C1653         ZCOEFF=0.0
C1654         M=-NMODE
C1655         DO 184 MM=1,KMODE
C1656         M=M+1
01657         CALL MINT(M,ZC(J),Z1(J),A3,A4)
01658         TCOEFF=TCOEFF+A3
C1659         ZCOEFF=ZCOEFF+A4
C1660  184    CONTINUE
C1661         J2=2*(IB-1)+1
C1662         J4=4*(IB-1)+1
C1664         TCOEFF=TCOEFF*(DH(J2)*T(J4)+DH(J2+1)*T(J4+1)+
C1665       1 DH(J2+2)*T(J4+2)+DH(J2+3)*T(J4+3))
C1666         ZCOEFF=ZCOEFF*(DH(J2)*TZ(J4)+DH(J2+1)*TZ(J4+1)+
C1667       1 DH(J2+2)*TZ(J4+2)+DH(J2+3)*TZ(J4+3))
01670         DO 185 I=1,NSA
C1680         IA=IS(I)
C1690         I2=2*(IA-1)+1
C1700         I4=4*(IA-1)+1
C1710         K=K+1
C1720         TAB(K)=TAB(K)*(DH(I2)*T(I4)+DH(I2+1)*T(I4+1)+
C1730       1 DH(I2+2)*T(I4+2)+DH(I2+3)*T(I4+3))/TCOEFF
C1740         ZAB(K)=ZAB(K)*(DH(I2)*TZ(I4)+DH(I2+1)*TZ(I4+1)+
C1750       1 DH(I2+2)*TZ(I4+2)+DH(I2+3)*TZ(I4+3))/ZCOEFF
01810         WRITE(6,202) I,J,TAB(K),ZAB(K)
C1820  202    FORMAT(I8,I9,2X,4G10.3)
C1830  185    CONTINUE
C1840         WRITE(6,201)
C1850  201    FORMAT(//,' WIRE-WIRE ANTENNA COUPLING',//,
C1860       1 '    JUNC A    JUNC B    COUPLING COEFF. (COMPLEX)',/)
C1865         K=0
C1870         DO 200 J=1,NJ
C1880         DO 200 I=1,NJ
C1885         K=K+1
C1910         WRITE(6,202) I,J,WAB(K)
C1920  200    CONTINUE
C2812         CALL PLOT(XPLOT,YPLOT,ISYM,NPLOT,'X','Y',101,51)
```

```
C2814         CALL PLOT(ZPLOT,YPLOT,ISYM,NPLOT,'Z','Y',101,51)
C2816         CALL PLOT(ZPLOT,XPLOT,ISYM,NPLOT,'Z','X',101,51)
C2818         STOP
C2820         END
C3370C ---------------------------------------------------------------
C3375         SUBROUTINE BOTIN
C3380C
C3385C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
C3390C
C3392         COMMON /PLOT1/ NPLOT,XPLOT(200),YPLOT(200),ZPLOT(200),ISYM(200)
C3395         COMMON /BOT1/ NMODE,NPT,NBAND
C3400         COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C3405         COMMON /BOT3/ D1(82),SV(82),CV(82)
C3410         COMMON /BOT5/ T(160),TP(160),TZ(160)
C3415         COMMON /BOT6/ IEDGE,IUNIF
C3420         READ(5,49) NMODE,NPT,NBAND
C3425         READ(5,49) NP
C3430 49      FORMAT(3I3)
C3435         WRITE(6,48) NMODE,NPT,NBAND,NP
C3440 48      FORMAT(32H     NMODE    NPT     NBAND      NP,/,4I8)
C3445         READ(5,53)(YB(I),I=1,NP)
C3450         READ(5,53)(XB(I),I=1,NP)
C3455    53   FORMAT(10F8.4)
C3460         WRITE(6,55)
C3465 55      FORMAT(/,3H YB)
C3470         WRITE(6,46)(YB(I),I=1,NP)
C3475    46   FORMAT(1X,10F8.4)
C3480         WRITE(6,56)
C3485 56      FORMAT(/,3H XB)
C3486         WRITE(6,46)(XB(I),I=1,NP)
C3488C PLOT THE BODY COORDINATES.
C3490         CALL PLOTB(XB,YB,NP,41)
C3492         READ(5,53) BL
C3494         WRITE(6,47) BL
C3496 47      FORMAT(//,21H HALF-LENGTH OF BOT =,F12.4)
C3500C UPDATE PLOT ARRAY.
C3501         DO 150 I=1,NP
C3502         NPLOT=NPLOT+1
C3503         XPLOT(NPLOT)=XB(I)
C3504         YPLOT(NPLOT)=YB(I)
C3505         ZPLOT(NPLOT)=BL
C3506         ISYM(NPLOT)='B'
C3507         NPLOT=NPLOT+1
C3508         XPLOT(NPLOT)=XB(I)
C3509         YPLOT(NPLOT)=YB(I)
C3510         ZPLOT(NPLOT)=-BL
C3511         ISYM(NPLOT)='B'
C3512 150     CONTINUE
C3520C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
C3525         IF(MOD(NP,2).NE.1) GO TO 980
C3530C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
C3535         IEDGE=1
```

```
C3540       IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
C3545       IEDGE=0
C3550       YB(NP+1)=YB(2)
C3555       XB(NP+1)=XB(2)
C3560       YB(NP+2)=YB(3)
C3565       XB(NP+2)=XB(3)
C3570       NP=NP+2
C3575       WRITE(6,66) NP
C3580  66   FORMAT(//,39H BOT GENERATING CURVE IS CLOSED.  NP = ,I3)
C3585  10   CONTINUE
C3590C COMPUTATION OF BODY SEGMENT PARAMETERS.
C3595       DO 57 I=2,NP
C3600       I2=I-1
C3605       RR1=YB(I)-YB(I2)
C3610       RR2=XB(I)-XB(I2)
C3615       DH(I2)=SQRT(RR1*RR1+RR2*RR2)
C3620       XB1(I2)=.5*(XB(I)+XB(I2))
C3625       YB1(I2)=.5*(YB(I)+YB(I2))
C3630       SV(I2)=RR1/DH(I2)
C3635       CV(I2)=RR2/DH(I2)
C3640  57   CONTINUE
C3645C CHECK IF BOT SEGMENTATION IS UNIFORM.
C3650       IUNIF=0
C3655       NP1=NP-1
C3660       DO 60 I=2,NP1
C3665       RR1=DH(I)/DH(1)
C3670       IF(RR1.LT.0.99 .OR. RR1.GT.1.01) GO TO 20
C3675  60   CONTINUE
C3680       IUNIF=1
C3685       WRITE(6,67)
C3690  67   FORMAT(//,' BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
C3695  20   CONTINUE
C3700C COMPUTATION OF TRIANGLE FUNCTIONS T.
C3705       NM=(NP-3)/2
C3710       DO 74 J=1,NM
C3715       J2=2*(J-1)+1
C3720       J3=J2+1
C3725       J4=J3+1
C3730       J5=J4+1
C3735       J6=4*(J-1)+1
C3740       J7=J6+1
C3745       J8=J7+1
C3750       J9=J8+1
C3755       DEL1=DH(J2)+DH(J3)
C3760       DEL2=DH(J4)+DH(J5)
C3765       TP(J6)=1./DEL1
C3770       TP(J7)=1./DEL1
C3775       TP(J8)=-1./DEL2
C3780       TP(J9)=-1./DEL2
C3785       T(J6)=DH(J2)/2./DEL1
C3790       T(J7)=(DH(J2)+DH(J3)/2.)/DEL1
C3795       T(J8)=(DH(J4)/2.+DH(J5))/DEL2
```

```
C3800       T(J9)=DH(J5)/2./DEL2
C3805    74 CONTINUE
C3810       NM4=N*4
C3815       DO 75 J=1,NM4
C3820  75    TZ(J)=T(J)
C3825       IF(IEDGE.EQ.0) GO TO 76
C3830       TZ(1)=2.-T(1)
C3835       TZ(2)=2.-T(2)
C3840       TZ(NM4-1)=2.-T(NM4-1)
C3845       TZ(NM4)=2.-T(NM4)
C3850  76    CONTINUE
C3855       RETURN
C3860  980   WRITE(6,981)
C3865  981   FORMAT(//,' **** ERROR IN BDT INPUT')
C3870       STOP
C3875       END
C3880C ******************************************************
C3882       SUBROUTINE PLOTB(X,Y,N,NR)
C3884C ******************************************************************
C3886C
C3888C WRITTEN 2/14/74   BY J. M. PUTNAM        DEPT 220      X23877
C3890C
C3892C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C3894C     N IS THE NUMBER OF POINTS TO BE PLOTTED.
C3896C     NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C3898C     NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C3900C        NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C3902C
C3904       REAL X(1),Y(1),HEAD(10)
C3906       INTEGER  LINE(101),BLANK,STAR,PLUS
C3908       DATA BLANK,STAR,PLUS /1H ,1H*,1H+/
C3910       NC=51
C3912       N10=(NC-1)/10
C3914       WRITE(6,500)
C3916  500   FORMAT(//,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
C3918       WRITE(6,504)
C3920       XMIN=X(1)
C3922       XMAX=X(1)
C3924       YMIN=Y(1)
C3926       YMAX=Y(1)
C3928       DO 6 I=1,N
C3930       IF(X(I).LT.XMIN) XMIN=X(I)
C3932       IF(X(I).GT.XMAX) XMAX=X(I)
C3934       IF(Y(I).LT.YMIN) YMIN=Y(I)
C3936       IF(Y(I).GT.YMAX) YMAX=Y(I)
C3938  6     CONTINUE
C3940       DEL=XMAX-XMIN
C3942       IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C3944       XMAX=XMIN+DEL
C3946       YMAX=YMIN+DEL
C3948       DO 5 I =1,N10
C3950       Z=I
```

```
C3952      5 HEAD(I)=(XMAX-XMIN)*Z/NIC+XMIN
C3954        DY=(YMAX-YMIN)/(NP-1)
C3956        IEDGE=1
C3958        IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
C3960        Z=YMAX+DY
C3962        YL=Z-DY/2.
C3964        DO 7 J=1,NP
C3966        DO 8 K=1,NC
C3968  8     LINE(K)=BLANK
C3970        Z=Z-DY
C3972        YU=YL
C3974        YL=Z-DY/2.
C3976        DO 9 I=1,N
C3978        IF(Y(I).GE.YU) GO TO 9
C3980        IF(Y(I).LT.YL) GO TO 9
C3982        K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
C3984        IF(K.GT.NC) K=NC
C3986        LINE(K)=STAR
C3988        IF(MOD(I,2).EQ.1) LINE(K)=PLUS
C3990        IF(IEDGE.EQ.0) GO TO 9
C3992        IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
C3994  9     CONTINUE
C3996        WRITE(6,508) Z,(LINE(K),K=1,NC)
C3998  7     CONTINUE
C4000        WRITE(6,504)
C4002        WRITE( 6,3002)
C4004        WRITE( 6,507) XMIN,(HEAD(I),I=1,N1))
C4006        RETURN
C4008   504 FORMAT ( 1X, 14(1H-), 14., 10(5H----.), 1H- )
C4010   507 FORMAT(10X,11(F10.4))
C4012   508 FORMAT (1X, F12.4,1X, 1HI, 51A1, 1HI )
C4014  3002 FORMAT(4X,7HYH / XH,4X,1HI,5(9X,1HI))
C4016        END
C4410C ------------------------------------------------------
C4420        SUBROUTINE CAPIN
C4430C
C4440C READ CAP INPUTS AND COMPUTE CAP ARRAYS.
C4450C
C4460        COMMON /PLOT1/ NPLOT,XPLOT(200),YPLOT(200),ZPLOT(200),ISYM(200)
C4470        COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C4480        COMMON /CAP1/ NC,XC,YC,ZC(2)
C4490        COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
C4500        COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
C4510        COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
C4520        COMMON /EDG1/ NE,ZE(2),ZBE(10)
C4530        COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
C4540        READ(5,1) NC,NPR,NE
C4550  1     FORMAT(3I3)
C4560        IF(NE.NE.0) NE=NC
C4570        WRITE(6,3) NC,NPR,NE
C4580  3     FORMAT(1H1   NC  NPR   NE,//,3I5,///)
C4590        IF(NC.EQ.0) RETURN
```

```
C4600       READ(5,2) XC,YC
C4610  2    FORMAT(10F8.4)
C4620       READ(5,2) (ZC(I),I=1,NC)
C4630       READ(5,2) (RHOC(I),I=1,NPR)
C4640       IF(NE.NE.0) READ(5,2) (ZE(I),I=1,NE)
C5100C UPDATE PLOT ARRAY.
C5102       DO 150 I=1,NC
C5104       NPLOT=NPLOT+1
C5106       XPLOT(NPLOT)=XC
C5108       YPLOT(NPLOT)=YC
C5110       ZPLOT(NPLOT)=ZC(I)
C5112       ISYM(NPLOT)='C'
C5114       IF(NE.EQ.0) GO TO 150
C5116       NPLOT=NPLOT+1
C5118       XPLOT(NPLOT)=XB(1)
C5120       YPLOT(NPLOT)=YB(1)
C5122       ZPLOT(NPLOT)=ZE(I)
C5124       ISYM(NPLOT)='E'
C5126  150  CONTINUE
C5144       WRITE(6,4)
C5146  4    FORMAT(3H  CAP       XC       YC       ZC       ZE,/)
C5148       DO 100 I=1,NC
C5149       IF(NE.EQ.0) WRITE(6,5) I,XC,YC,ZC(I)
C5150       IF(NE.NE.0) WRITE(6,5) I,XC,YC,ZC(I),ZE(I)
C5151  100  CONTINUE
C5152  5    FORMAT(I4,4X,4F8.4)
C5154       WRITE(6,6)
C5156  6    FORMAT(//,5H RHOC)
C5158       WRITE(6,7) (RHOC(I),I=1,NPR)
C5160  7    FORMAT(1X,10F8.4)
C5162       IF(MCD(NPR,2).NE.1) GO TO 990
C5164       DO 120 I=2,NPR
C5166  120  IF(RHOC(I).LE.RHOC(I-1)) GO TO 980
C5168C COMPUTATION OF CAP SECTOR PARAMETERS.
C5170       DO 47 I=1,NP
C5172       RC(I)=SQRT((YB(I)-YC)**2+(XB(I)-XC)**2)
C5174  47   CONTINUE
C5176       DO 57 I=2,NP
C5178       I2=I-1
C5180       RR1=YB1(I2)-YC
C5182       RR2=XB1(I2)-XC
C5184       RC1(I2)=SQRT(RR1*RR1+RR2*RR2)
C5186       AC(I2)=ABS((XB(I)-XC)*(YB(I)+YC)+(XB(I2)-XB(I))*
C5188      1 (YB(I2)+YB(I))+(XC-XB(I2))*(YC+YB(I2)))/2.
C5190       SPC(I2)=RR1/RC1(I2)
C5192       CPC(I2)=RR2/RC1(I2)
C5194  57   CONTINUE
C5196       DO 67 I=2,NPR
C5198       I2=I-1
C5200       RHOC1(I2)=(RHOC(I)+RHOC(I2))/2.
C5202       DRHOC(I2)=RHOC(I)-RHOC(I2)
C5204  67   CONTINUE
```

```
C5206C COMPUTATION OF CAP TRIANGLE FUNCTIONS.
C5208        LC=(NPR-3)/2
C5210        DO 74 J=1,LC
C5212        J2=2*(J-1)+1
C5214        J3=J2+1
C5216        J4=J3+1
C5218        J5=J4+1
C5220        J6=4*(J-1)+1
C5222        J7=J6+1
C5224        J8=J7+1
C5226        J9=J8+1
C5228        DEL1=DRHOC(J2)+DRHOC(J3)
C5230        DEL2=DRHOC(J4)+DRHOC(J5)
C5232        TPCR(J6)=1./DEL1
C5234        TPCR(J7)=1./DEL1
C5236        TPCR(J8)=-1./DEL2
C5238        TPCR(J9)=-1./DEL2
C5240        TCR(J6)=DRHOC(J2)/2./DEL1
C5242        TCR(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
C5244        TCR(J8)=(DRHOC(J4)/2.+DRHOC(J5))/DEL2
C5246        TCR(J9)=DRHOC(J5)/2./DEL2
C5248     74 CONTINUE
C5250        LC4=LC*4
C5252        DO 75 I=1,LC4
C5254        TCT(I)=TCR(I)
C5256        TPCT(I)=TPCR(I)
C5258  75    CONTINUE
C5260        TCT(LC4-1)=2.-TCT(LC4-1)
C5262        TCT(LC4)=2.-TCT(LC4)
C5264        TPCT(LC4-1)=-TPCT(LC4-1)
C5266        TPCT(LC4)=-TPCT(LC4)
C5268        IF(PHOC(1).EQ.0.0) GO TO 76
C5270        TCT(1)=2.-TCT(1)
C5272        TCT(2)=2.-TCT(2)
C5274        TPCT(1)=-TPCT(1)
C5276        TPCT(2)=-TPCT(2)
C5278  76    CONTINUE
C5279        IF(NE.EQ.0) RETURN
C5280C COMPUTATION OF EDGE HALF TRIANGLE FUNCTIONS.
C5282        DO 80 IC=1,NC
C5284        J2=NPR-2
C5286        J3=J2+1
C5288        J6=2*(IC-1)+1
C5290        J7=J6+1
C5292        DEL1=DRHOC(J2)+DRHOC(J3)
C5294        TPCE(J6)=1./DEL1
C5296        TPCE(J7)=1./DEL1
C5298        TCE(J6)=DRHOC(J2)/2./DEL1
C5300        TCE(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
C5302        DEL2=ZE(IC)-ZC(IC)
C5304        ZBE(J6)=ZC(IC)+0.25*DEL2
C5306        ZBE(J7)=ZC(IC)+0.75*DEL2
```

```
      IF(DEL2.LT.0.0) GO TO 78
C EDGE IS AT Z=-L.
      TPBE(J6)=-1./DEL2
      TPBE(J7)=-1./DEL2
      TBE(J6)=0.75
      TBE(J7)=0.25
      GO TO 80
78    CONTINUE
C EDGE IS AT Z=+L.
      TPBE(J6)=1./DEL2
      TPBE(J7)=1./DEL2
      TBE(J6)=-0.75
      TBE(J7)=-0.25
80    CONTINUE
      RETURN
980   WRITE(6,981)
981   FORMAT(//,' **** ERROR IN CAP INPUT')
      STOP
      END
C-----------------------------------------------------
      SUBROUTINE WIREIN
C
C READ WIRE COORDINATES AND COMPUTE WIRE SEGMENT ARRAYS.
C
      COMMON /PLOT1/ NPLOT,XPLOT(200),YPLOT(200),ZPLOT(200),ISYM(200)
      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
     1 XW1(100),YW1(100),ZW1(100)
      COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
      COMMON /WIRE3/ NW,INDW(6),RADW(5)
      COMMON /WIRE4/ LW,IW(195),IPW(196),INDIW(49)
      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
      COMMON /JUNC2/ IJ(20),IPJ(20),INDIJ(10)
      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
      COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
      NW=0
      NJ=0
      LW=0
      READ(5,51) NW,NPW,NJ
51    FORMAT(3I3)
      WRITE(6,1) NW,NPW,NJ
1     FORMAT(24H1      NW      NPW      NJ,/,318)
      IF(NW.EQ.0) RETURN
      READ(5,53) (XW(I),I=1,NPW)
      READ(5,53) (YW(I),I=1,NPW)
      READ(5,53) (ZW(I),I=1,NPW)
53    FORMAT(10F8.4)
      READ(5,52) (INDW(I),I=1,NW)
52    FORMAT(10I8)
      INDW(NW+1)=NPW+1
      READ(5,53) (RADW(I),I=1,NW)
```

```
      IF(NJ.EQ.0) GO TO 50
C NOTE, INDJW MUST BE MONOTONIC INCREASING.
      READ(5,52) (INDJW(I),I=1,NJ)
      READ(5,53) (RADJ(I),I=1,NJ)
      READ(5,53) (UXJ(I),I=1,NJ)
      READ(5,53) (UYJ(I),I=1,NJ)
      READ(5,53) (UZJ(I),I=1,NJ)
50    IERW=0
      IERJ=0
      WRITE(6,61)
61    FORMAT(/,22X,'WIRE COORDINATES',20X,'JUNCTION PARAMETERS',//,
     1 26X,'IW',4X,'XW',6X,'YW',6X,'ZW',
     2 10X,'IJ',4X,'RADJ',5X,'UXJ',5X,'UYJ',5X,'UZJ')
      IJ=1
C THIS LOOP LISTS WIRE/JUNCTION POINTS, WHILE CHECKING THE FOLLOWING:
C   1) EACH WIRE MUST CONTAIN AN ODD NUMBER OF POINTS.
C   2) EACH JUNCTION MUST EITHER START OR TERMINATE A WIRE.
C   3) CHECK THAT ALL JUNCTION POINTS ARE FOUND.
      DO 100 IW=1,NW
      WRITE(6,62) IW,RADW(IW)
62    FORMAT(2X,'WIRE',I3,' RADW=',F8.4)
      I1=INDW(IW)
      I2=INDW(IW+1)-1
C CHECK FOR AN ODD NUMBER OF POINTS ON WIRE IW.
      IF(MOD(I2-I1+1,2).NE.1) IERW=1
      DO 90 I=I1,I2
      WRITE(6,63) I,XW(I),YW(I),ZW(I)
63    FORMAT(25X,I3,3F8.4)
C UPDATE PLOT ARRAY.
      NPLOT=NPLOT+1
      XPLOT(NPLOT)=XW(I)
      YPLOT(NPLOT)=YW(I)
      ZPLOT(NPLOT)=ZW(I)
      ISYM(NPLOT)='W'
C CHECK IF WIRE POINT I IS A JUNCTION POINT.
      IF(IJ.GT.NJ) GO TO 90
      IF(INDJW(IJ).NE.I) GO TO 90
C CHECK THAT JUNCTION POINT IJ IS AT THE START OR END OF WIRE.
      IF(I.NE.I1 .AND. I.NE.I2) IERJ=1
      WRITE(6,64) IJ,RADJ(IJ),UXJ(IJ),UYJ(IJ),UZJ(IJ)
64    FORMAT('+',58X,I3,4F8.4)
C UPDATE PLOT ARRAY.
      ISYM(NPLOT)='J'
C DETERMINE DIRECTION IN WHICH WIRE LEAVES JUNCTION POINT.
      INDTJ(IJ)=1
      IF(I.EQ.I2) INDTJ(IJ)=-1
C COMPUTE JUNCTION PARAMETERS.
      RADJ(IJ)=RADW(IW)
      XJ(IJ)=XW(I)
      YJ(IJ)=YW(I)
      ZJ(IJ)=ZW(I)
      IJ=IJ+1
```

```
C5890 90    CONTINUE
C5895 100   CONTINUE
C5900C CHECK FOR WIRE OR JUNCTION INPUT ERRORS.
C5905       IF(IERW.NE.0) GO TO 980
C5910       IF(IJ-1.NE.NJ .OR. IERJ.NE.0) GO TO 990
C5915C COMPUTATION OF WIRE SEGMENT PARAMETERS.
C5920       DO 57 I=2,NPW
C5925       I2=I-1
C5930       DXW(I2)=XW(I)-XW(I2)
C5935       DYW(I2)=YW(I)-YW(I2)
C5940       DZW(I2)=ZW(I)-ZW(I2)
C5945       DHW(I2)=SQRT(DXW(I2)**2+DYW(I2)**2+DZW(I2)**2)
C5950       XW1(I2)=0.5*(XW(I)+XW(I2))
C5955       YW1(I2)=0.5*(YW(I)+YW(I2))
C5960       ZW1(I2)=0.5*(ZW(I)+ZW(I2))
C5965    57 CONTINUE
C5970C COMPUTATION OF WIRE TRIANGLE FUNCTIONS TW.
C5975       LW=0
C5980       DO 75 IW=1,NW
C5985       I1=INDW(IW)
C5990       I2=INDW(IW+1)-1
C5995       LW1=(I2-I1-2)/2
C6000       DO 74 J=1,LW1
C6005       LW=LW+1
C6010       J2=2*(J-1)+I1
C6015       J3=J2+1
C6020       J4=J3+1
C6025       J5=J4+1
C6030       J6=4*(LW-1)+1
C6035       J7=J6+1
C6040       J8=J7+1
C6045       J9=J8+1
C6050       INDTW(LW)=J2
C6055       DEL1=DHW(J2)+DHW(J3)
C6060       DEL2=DHW(J4)+DHW(J5)
C6065       TPW(J6)=1./DEL1
C6070       TPW(J7)=1./DEL1
C6075       TPW(J8)=-1./DEL2
C6080       TPW(J9)=-1./DEL2
C6085       TW(J6)=DHW(J2)/2./DEL1
C6090       TW(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C6095       TW(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C6100       TW(J9)=DHW(J5)/2./DEL2
C6105 74    CONTINUE
C6110 75    CONTINUE
C6115C NOTE, LW=(NPW-NW)/2-NW
C6120       IF(NJ.EQ.0) RETURN
C6125C COMPUTATION OF JUNCTION HALF TRIANGLE FUNCTIONS TJ.
C6130       DO 85 IJ=1,NJ
C6135       IF(INDTJ(IJ).GT.0) GO TO 80
C6140C JUNCTION IS AT THE END OF A WIRE.
C6145       J2=INDJW(IJ)-2
```

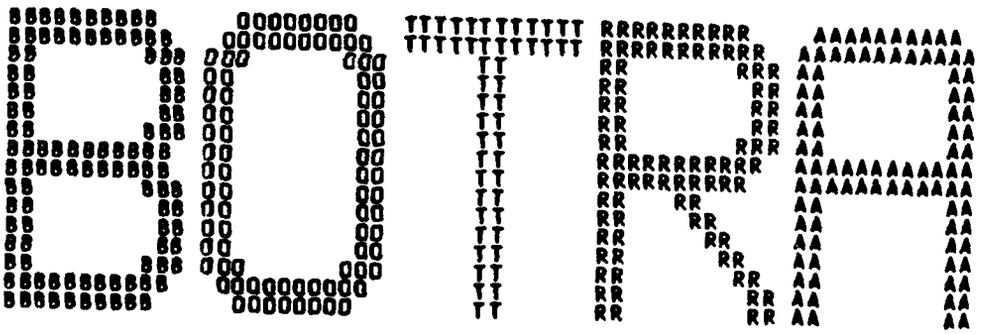
```
C6150         J3=J2+1
C6155         J6=2*(IJ-1)+1
C6160         J7=J6+1
C6165         INDTJ(IJ)=J2
C6170         DEL1=-DHW(J2)-DHW(J3)
C6175         TPJ(J6)=1./DEL1
C6180         TPJ(J7)=1./DEL1
C6185         TJ(J6)=DHW(J2)/2./DEL1
C6190         TJ(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C6195         GO TO 85
C6200C JUNCTION IS AT THE START OF A WIRE.
C6205 80      J4=INDJW(IJ)
C6210         J5=J4+1
C6215         J8=2*(IJ-1)+1
C6220         J9=J8+1
C6225         INDTJ(IJ)=J4
C6230         DEL2=DHW(J4)+DHW(J5)
C6235         TPJ(J8)=-1./DEL2
C6240         TPJ(J9)=-1./DEL2
C6245         TJ(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C6250         TJ(J9)=DHW(J5)/2./DEL2
C6255 85      CONTINUE
C6260C COMPUTE UNIT VECTORS FOR JUNCTION(DISK).
C6265         DO 95 IJ=1,NJ
C6270C NORMAL UNIT VECTOR (UJ).
C6275         RR=SQRT(UXJ(IJ)**2+UYJ(IJ)**2+UZJ(IJ)**2)
C6280         UXJ(IJ)=UXJ(IJ)/RR
C6285         UYJ(IJ)=UYJ(IJ)/RR
C6290         UZJ(IJ)=UZJ(IJ)/RR
C6295C FIND 2 ORTHOGONAL UNIT VECTORS IN THE PLANE OF THE DISK (UJ1 & UJ2).
C6300         UXJ1(IJ)=0.0
C6305         UYJ1(IJ)=0.0
C6310         UZJ1(IJ)=0.0
C6315         UXJ2(IJ)=0.0
C6320         UYJ2(IJ)=0.0
C6325         UZJ2(IJ)=0.0
C6330         IF(UXJ(IJ).EQ.0.0) GO TO 91
C6335C FIND INTERSECTION WITH X-Y PLANE.
C6340         UYJ1(IJ)=1.0
C6345         UXJ1(IJ)=-UYJ(IJ)/UXJ(IJ)
C6350C FIND INTERSECTION WITH X-Z PLANE.
C6355         UZJ2(IJ)=1.0
C6360         UXJ2(IJ)=-UZJ(IJ)/UXJ(IJ)
C6365         GO TO 94
C6370 91      IF(UYJ(IJ).EQ.0.0) GO TO 92
C6375C FIND INTERSECTION WITH Y-Z PLANE.
C6380         UZJ1(IJ)=1.0
C6385         UYJ1(IJ)=-UZJ(IJ)/UYJ(IJ)
C6390C FIND INTERSECTION WITH X-Y PLANE.
C6395         UXJ2(IJ)=1.0
C6400         UYJ2(IJ)=-UXJ(IJ)/UYJ(IJ)
C6405         GO TO 94
```

```
 92    IF(UZJ(IJ).EQ.0.0) GO TO 94
C FIND INTERSECTION WITH X-Z PLANE.
       UXJ1(IJ)=1.0
       UZJ1(IJ)=-UXJ(IJ)/UZJ(IJ)
C FIND INTERSECTION WITH Y-Z PLANE.
       UYJ2(IJ)=1.0
       UZJ2(IJ)=-UYJ(IJ)/UZJ(IJ)
 94    CONTINUE
       RR=SQRT(UXJ1(IJ)**2+UYJ1(IJ)**2+UZJ1(IJ)**2)
       UXJ1(IJ)=UXJ1(IJ)/RR
       UYJ1(IJ)=UYJ1(IJ)/RR
       UZJ1(IJ)=UZJ1(IJ)/RR
       RR=SQRT(UXJ2(IJ)**2+UYJ2(IJ)**2+UZJ2(IJ)**2)
       UXJ2(IJ)=UXJ2(IJ)/RR
       UYJ2(IJ)=UYJ2(IJ)/RR
       UZJ2(IJ)=UZJ2(IJ)/RR
 95    CONTINUE
       RETURN
 980   WRITE(6,981)
 981   FORMAT(//,' **** ERROR IN WIRE INPUT')
       STOP
 990   WRITE(6,991)
 991   FORMAT(//,' **** ERROR IN JUNCTION INPUT')
       STOP
       END
C-----------------------------------------------------------
       SUBROUTINE SLOTIN
C
C READ SLOT ANTENNA INPUTS.
C
       COMPLEX EO
       COMMON /PLOT1/ NPLOT,XPLOT(200),YPLOT(200),ZPLOT(200),ISYM(200)
       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
       COMMON /SLOT1/ NSA,IS(20),ZO(20),Z1(20)
       COMMON /SLOT2/ EO(20),TEXC(20),ZEXC(20)
       READ(5,16) NSA
 16    FORMAT(I3)
       WRITE(6,3) NSA
 3     FORMAT(//,26H1NUMBER OF SLOT ANTENNAS =,I3)
       IF(NSA.EQ.0) RETURN
       READ(5,54) (IS(K),K=1,NSA)
 54    FORMAT(10I8)
C NOTE -L <= ZO(K) < Z1(K) <= L,
       READ(5,53) (ZO(K),K=1,NSA)
 53    FORMAT(10F8.4)
       READ(5,53) (Z1(K),K=1,NSA)
       READ(5,53) (EO(K),K=1,NSA)
C T-EXCITATION.
       READ(5,53) (TEXC(K),K=1,NSA)
C Z-EXCITATION.
       READ(5,53) (ZEXC(K),K=1,NSA)
       WRITE(6,4) ((K,IS(K),ZO(K),Z1(K),EO(K),TEXC(K),ZEXC(K)),K=1,NSA)
```

```
C7900 4     FORMAT(///,50H ANTENNA NO.   IS    ZO          Z1                EO,
C7910      1 11X,14HTEXC       ZEXC,//,(I9,I8,6F10.4))
C7911C UPDATE PLOT ARRAY.
C7912       DO 150 I=1,NSA
C7913       NPLOT=NPLOT+1
C7914       IND=IS(I)
C7915       XPLOT(NPLOT)=XB(2*IND+1)
C7916       YPLOT(NPLOT)=YB(2*IND+1)
C7917       ZPLOT(NPLOT)=ZO(I)
C7918       ISYM(NPLOT)='S'
C7919       NPLOT=NPLOT+1
C7920       XPLOT(NPLOT)=XB(2*IND+1)
C7921       YPLOT(NPLOT)=YB(2*IND+1)
C7922       ZPLOT(NPLOT)=Z1(I)
C7923       ISYM(NPLOT)='S'
C7924 150   CONTINUE
C7925       RETURN
C7930       END
C7940C ------------------------------------------------------
C7950       SUBROUTINE ORDER(MORD)
C7960C DETERMINE THE ORDER IN WHICH THE Y MATRIX WAS GENERATED, AND
C7970C CHECK FOR DIMENSIONAL CONSISTENCY BETWEEN THE Y MATRIX DATA FILE,
C7980C AND THE INPUT DATA FILE.
C7990       DIMENSION MORD(2)
C8000       COMMON /BOT1/ NMODE,NPT,NBAND
C8010       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C8020       COMMON /CAP1/ NC,XC,YC,ZC(2)
C8025       COMMON /CAP2/ NPR,PHDC(21),RHDC1(20),DRHDC(20)
C8030       COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
C8032      1 XW1(100),YW1(100),ZW1(100)
C8034       COMMON /WIRE3/ NW,INDW(6),RADW(5)
C8040       COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
C8042       COMMON /EDG1/ NE,ZE(2),ZBE(10)
C8050       READ(1) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
C8052       IF(NMODE.NE.NMODE1) GO TO 500
C8054       IF(NP.NE.NP1) GO TO 500
C8056       IF(NC.NE.NC1) GO TO 500
C8058       IF(NPR.NE.NPR1) GO TO 500
C8060       IF(NE.NE.NE1) GO TO 500
C8062       IF(NW.NE.NW1) GO TO 500
C8064       IF(NPW.NE.NPW1) GO TO 500
C8066       IF(NJ.NE.NJ1) GO TO 500
C8120C MORD SPECIFIES THE ORDER IN WHICH CAPS AND/OR WIRES HAVE BEEN ADDED
C8130C TO THE BOT.  INDEX 1 REFERS TO CAPS AND INDEX 2 REFERS TO WIRES.
C8140C   MORD(I)=0 IF I HAS NOT BEEN ADDED
C8150C   MORD(I)=N IF I HAS BEEN ADDED TO THE SYSTEM MATRIX, AND
C8160C   IS LOCATED AT PSEUDO MODE NUMBER N.
C8170       READ(1) MORD
C8220       WRITE(6,4) MORD
C8230 4     FORMAT(' THE Y MATRIX CONTAINS THE FOLLOWING ADDITIONS ( 0 IF ',
C8240      1 ' NOT INCLUDED, OR CORRESPONDING MODE NUMBER IF PRESENT):',//,
C8250      2 10X,'CAPS',I4,/,10X,'WIRES',I3)
```

```
      RETURN
  500 WRITE(6,1)
    1 FORMAT(//,' *** ERROR ***   INPUT PARAMETERS DO NOT CHECK WITH',
     1 ' THE PARAMETERS IN THE Y MATRIX FILE',//,
     2 6X,'  NMODE   NP    NC   NPR    NE    NW   NPW    NJ')
      WRITE(6,2) NMODE,NP,NC,NPR,NE,NW,NPW,NJ
    2 FORMAT(' INPUT ',8I6)
      WRITE(6,3) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
    3 FORMAT(' Y FILE',8I6)
      STOP
      END
C-----------------------------------------------------------------
      SUBROUTINE MINT(M,Z0,Z1,TINT,ZINT)
C INTEGRATE T AND Z EXPANSION FUNCTIONS OVER SLOT.
      COMPLEX TINT,ZINT
      COMPLEX U
      COMMON /BOT2/ NP,BL,YB(83),XB(83),Y91(82),XB1(82)
      DATA PI,U /3.14159265,(0.,1.)/
      IF(M.EQ.0) TINT=Z1-Z0
      IF(M.NE.0) TINT=-U*BL/M/PI*(CEXP(U*M*PI*Z1/BL)-
     1 CEXP(U*M*PI*Z0/BL))
      ZINT=TINT-(-1)**M*(Z1-Z0)
      RETURN
      END
C-----------------------------------------------------------------
      SUBROUTINE PLOT(X,Y,ISYM,N,LX,LY,NC,NR)
C-----------------------------------------------------------------
C
C WRITTEN 2/14/74   BY J. M. PUTNAM        DEPT 220      X23877
C
C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C    N IS THE NUMBER OF POINTS TO BE PLOTTED.
C    NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C    NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C       NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C
      DIMENSION X(1),Y(1),ISYM(1),HEAD(10)
      INTEGER  LINE(102),BLANK,PLUS
      DATA BLANK,PLUS /1H ,1H+/
      NC1=NC+1
      WRITE(6,501)
  501 FORMAT(1H1)
      N10=(NC-1)/10
      XMIN=X(1)
      XMAX=X(1)
      YMIN=Y(1)
      YMAX=Y(1)
      DO 6 I=1,N
      IF(X(I).LT.XMIN) XMIN=X(I)
      IF(X(I).GT.XMAX) XMAX=X(I)
      IF(Y(I).LT.YMIN) YMIN=Y(I)
      IF(Y(I).GT.YMAX) YMAX=Y(I)
```

```
16770  6     CONTINUE
16780        DELX=(XMAX-XMIN)/(NC-1)
16790        DELY=(YMAX-YMIN)/(NR-1)
16800        DO 5 I=1,N10
16810  5     HEAD(I)=XMIN+10*I*DELX
16820        DO 11 K=1,NC1
16830        LINE(K)='-'
16840        IF(MOD(K,5).EQ.1) LINE(K)='.'
16850  11    CONTINUE
16860        WRITE(6,504) (LINE(K),K=1,NC1)
16870        YU=YMAX+DELY/2.
16880        YL=YMAX-DELY/2.
16890        DO 7 J=1,NR
16900        DO 8 K=1,NC
16910  8     LINE(K)=BLANK
16920        LINE(NC1)='I'
16930        DO 9 I=1,N
16940        IF(Y(I).GE.YU) GO TO 9
16950        IF(Y(I).LT.YL) GO TO 9
16960        K=(X(I)-XMIN)/DELX+1.5
16970        IF(LINE(K).NE.BLANK) LINE(K)=PLUS
16980        IF(LINE(K).EQ.BLANK) LINE(K)=ISYM(I)
16990  9     CONTINUE
17000        YM=(YU+YL)/2.
17010        WRITE(6,508) YM,(LINE(K),K=1,NC1)
17020        YU=YL
17030        YL=YL-DELY
17040  7     CONTINUE
17050        DO 12 K=1,NC1
17060        LINE(K)='-'
17070        IF(MOD(K,5).EQ.1) LINE(K)='.'
17080  12    CONTINUE
17090        WRITE(6,504) (LINE(K),K=1,NC1)
17100        DO 13 K=1,NC1
17110        LINE(K)=BLANK
17120        IF(MOD(K,10).EQ.1) LINE(K)='I'
17130  13    CONTINUE
17140        WRITE( 6,3002) LY,LX,(LINE(K),K=1,NC1)
17150        WRITE( 6,507) XMIN,(HEAD(I),I=1,N10)
17160        WRITE(6,502) N
17170  502   FORMAT(/,2X,I4,15H POINTS PLOTTED)
17180        RETURN
17190  504   FORMAT(1X,14(1H-),102A1)
17200   507  FORMAT(1CX,11(F10.4))
17210   508  FORMAT (1X, F12.4,1X, 1HI, 102A1 )
17220  3002  FORMAT(4X,A1,5H   /  ,A1,4X,102A1)
17230        END
1724C/EOR
```


BOTRA

```
C0100BOTR,P30,T100,CM150000,STCX3.
C0110ACCOUNT,MC3265,BOT2A.
C0120BANNERS(OUTPUT) *J. PUTNAM*DEPT 220*BLD 113-4**
C0130ATTACH(INFIL=YSSX)
C0140FTN(R=0,OPT=1)
C0164LDSET(PRESET=INDEF)
C0170LGO.
C0190/EOR
C0200       PROGRAM BOTRA(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C0210      1 INFIL,TAPE1=INFIL)
C0220C
C0230C BOTRA IS THE BOT RADIATION CODE.
C0240C
C0250C UNIT 5 IS THE CARD READER.
C0260C UNIT 6 IS THE LINE PRINTER.
C0270C UNIT 1 IS A DISK FILE CONTAINING THE Y MATRIX.
C0280C
C0285       COMPLEX U
C0290       COMPLEX VB(82),VW(59)
C0300       COMPLEX GT(73),GP(73)
C0305       COMPLEX CB(1680),CW(59),CC(96)
C0310       COMPLEX ESC(3),HSC(3)
C0320       COMPLEX Y(2304)
C0325       COMPLEX RBT(82),RBP(82),RWT(59),RWP(59),RCT(96),RCP(96)
C0326       COMMON /PLOT1/ NPLOT,XPLOT(200),YPLOT(200),ZPLOT(200),ISYM(200)
00327       COMMON /WAVE/ BK
C0330       COMMON /BOT1/ NMODE,NPT,NBAND
C0340       COMMON /BOT2/ NP,BL,YB(83),XB(83),YBL(82),XBL(82)
C0350       COMMON /BOT3/ DH(82),SV(82),CV(82)
00370       COMMON /BOT5/ T(160),TP(160),TZ(160)
C0380       COMMON /BOT6/ IEDGE,IUNIF
C0381       COMMON /SLOT1/ NSA,IS(20),Z0(20),Z1(20)
C0382       COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
C0384      1 XW1(100),YW1(100),ZW1(100)
C0385       COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
C0386       COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
C0387       COMMON /CAP1/ NC,XC,YC,ZC(2)
C0388       COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
C0389       COMMON /EDG1/ NE,ZE(2),ZBE(10)
C0390       COMMON /INT/ M,RHO2,ZP
00400       DIMENSION THR(73),PHIR(73)
C0410       DIMENSION ANG(6),IPLANE(6),ANG1(6),ANG2(6)
C0420       DIMENSION MORD(2)
C0425       NPLOT=0
C0430       U=(0.,1.)
C0440       ETA=376.707
00450       PI=3.14159265
C0460       DTOR=PI/180.
00462       READ(5,1) BK
C0463 1     FORMAT(E15.7)
C0464       WRITE(6,2) BK
00465 2     FORMAT(9H1      BK,/,E15.7)
```

```
C0470C
C0480       CALL BOTIN
C0490C
C0500       CALL CAPIN
C0505       NCV=C
C0510C
C0520       CALL WIREIN
C0522C
C0540       CALL SLOTIN
C0550C
C0552       IF(NPW.NE.C) CALL VWIRE(NWJV,VW)
00554C
C0560C READ INPUTS, DESCRIBING THE RADIATION PLANES.
C0570       READ(5,44) NANG,NT
00580  44   FORMAT(2I3)
C0590       READ(5,53) (ANG(I),I=1,NANG)
C0600  53   FORMAT(10F8.4)
00610C  IPLANE = 1 FOR PHI FIXED.
C0620C          2 FOR THETA FIXED.
00630       READ(5,54) (IPLANE(I),I=1,NANG)
C0640  54   FORMAT(10I8)
C0642       READ(5,53) (ANG1(I),I=1,NANG)
C0644       READ(5,53) (ANG2(I),I=1,NANG)
00650       WRITE(6,3) NANG,NT
00660  3    FORMAT(//,25H NUMBER OF FIXED ANGLES =,I3,//
C0670      1 35H NUMBER OF ANGLES PER FIXED ANGLE =,I4)
C0680       WRITE(6,4) (ANG(I),IPLANE(I),ANG1(I),ANG2(I),I=1,NANG)
C0690  4    FORMAT(//,45H FIXED ANGLE    CODE    VARIABLE ANGLE RANGE,
C0700      1 //,(F8.1,9X,I2,6X,F8.1,3H - ,F8.1))
C0710       LS=NP-3
C0712       NM=LS/2
C0714       NP1=NM+1
C0716       NM4=NM*4
C0720       LSS=LS*LS
C0722       LR=(NPR-3)/2
C0724       LC=NC*NM*LR
C0726       LC2=LC*2
C0728       LE=NE*NM
C0730       LSC=LS*(LC2+LE)
C0735       LWJ=LW+NJ
C0740       LSW=LS*LWJ
00750       LCC=(LC2+LE)**2
C0760       LCW=(LC2+LE)*LWJ
C0770       LWW=LWJ**2
00810       BKL=BK*BL
C0830       KMODE=2*NMODE-1
C0840       CALL ZERO(LS*KMODE,CR)
C0860       IF(NC.NE.0) CALL ZERO(LC2+LE,CC)
C0870       IF(NPW.NE.C) CALL ZERO(LWJ,CW)
C0880       CALL CODER(MODE)
C0890C CALCULATE CURRENTS, EQ. 10.
C0900       NUMB=0
```

```
C0910  50    CONTINUE
CC920        READ(1) M,N
C0930        IF(ECF(1)) 37,60
C0940  60    NUMB=NUMB+1
C0942        IC=(P+NMODE-1)*LS+1
CC95CC DETERMINE SUBMATRIX TYPE, AND CALCULATE CURRENTS.
00960        IF(N.LT.NMODE) GO TO 61
CC970        IF(N.EQ.MORD(1)) GO TO 62
C0980        IF(N.EQ.MORD(2)) GO TO 63
C0990        GO TO 50
CC992  61    IF(NSA.EQ.0) GO TO 80
CC994        CALL VBOT(N,VB)
C1010        IF(M.LT.NMODE) GO TO 71
C1020        IF(M.EQ.MORD(1)) GO TO 72
C1030        IF(M.EQ.MORD(2)) GO TO 73
C1040        GO TO 50
C1042  62    IF(NCV.EQ.0) GO TO 80
C1050        IF(M.LT.NMODE) GO TO 74
C1060        IF(M.EQ.MORD(1)) GO TO 75
C1070        IF(M.EQ.MORD(2)) GO TO 76
C1080        GO TO 50
C1082  63    IF(NWJV.EQ.0) GO TO 80
C1090        IF(M.LT.NMODE) GO TO 77
01100        IF(M.EQ.MORD(1)) GO TO 78
01110        IF(M.EQ.MORD(2)) GO TO 79
C1120        GO TO 50
01130C BOT-BOT.
C1140  71    READ(1) (Y(I),I=1,LSS)
C1170        CALL MULTYV(LS,LS,Y,VB,CB(IC))
C1180        GO TO 50
01190C CAP-BOT.
01200  72    READ(1) (Y(I),I=1,LSC)
C1210        CALL MULTYV(LC2+LE,LS,Y,VB,CC)
C1220        GO TO 50
01230C WIRE-BOT.
C1240  73    READ(1) (Y(I),I=1,LSW)
C1250        CALL MULTYV(LWJ,LS,Y,VB,CW)
C1260        GO TO 50
C1270C BOT-CAP.
01280  74    READ(1) (Y(I),I=1,LSC)
01290        CALL MULTYV(LS,LC2+LE,Y,VC,CB(IC))
C1300        GO TO 50
01310C CAP-CAP.
C1320  75    READ(1) (Y(I),I=1,LCC)
C1330        CALL MULTYV(LC2+LE,LC2+LE,Y,VC,CC)
C1340        GO TO 50
01350C WIRE-CAP.
01360  76    READ(1) (Y(I),I=1,LCW)
C1370        CALL MULTYV(LWJ,LC2+LE,Y,VC,CW)
C1380        GO TO 50
C1390C BOT-WIRE.
01400  77    READ(1) (Y(I),I=1,LSW)
```

```
01410       CALL MULTYV(LS,LWJ,Y,VW,CB(IC))
01420       GO TO 50
01430C CAP-WIRE.
01440 78    READ(1) (Y(I),I=1,LCW)
01450       CALL MULTYV(LC2+LE,LWJ,Y,VW,CC)
01460       GO TO 50
01470C WIRE-WIRE.
01480 79    READ(1) (Y(I),I=1,LWW)
01490       CALL MULTYV(LWJ,LWJ,Y,VW,CW)
01500       GO TO 50
01542 80    READ(1) Y(1)
01544       GO TO 50
01550C ALL Y SUBMATRICES HAVE BEEN READ.
01560 37    CONTINUE
01562       WRITE(6,114) NUMB
01564 114   FORMAT(///,1X,I4,17H SUBMATRICES READ)
01570C CALCULATE PC.
01580       PC=0.0
01581       IF(NSA.EQ.0) GO TO 85
01582       DO 82 MM=1,KMODE
01583       M=-NPCOE+MM
01584       IC=(M+NMODE-1)*LS+1
01585       CALL VBOT(M,VB)
01586       CALL MULTVC(LS,VB,CB(IC),PO)
01587 82    CONTINUE
01588 85    CONTINUE
01589       IF(LC.NE.0 .AND. NCV.NE.0) CALL MULTVC(LC2+LE,VC,CC,PO)
01590       IF(NPW.NE.0 .AND. NWJV.NE.0) CALL MULTVC(LWJ,VW,CW,PO)
01591       WRITE(6,121) DBCON(PO)
01600 121   FORMAT(///,18H TOTAL POWER(DB) =,F8.2)
01630C *** END CALCULATION OF CURRENTS ***
01640       P1=8K*8K*ETA/4./PI/PO
01650       DO 90 IANG=1,NANG
01660C INITIALIZE ARRAYS, AND SET THE RADIATION ANGLES.
01670       IPLAN=IPLANE(IANG)
01675       DT=(ANG2(IANG)-ANG1(IANG))/(NT-1)
01680       DO 100 K=1,NT
01690       GT(K)=0.0
01700       GP(K)=0.0
01710       PHIR(K)=ANG(IANG)*(2-IPLAN)+(ANG1(IANG)+DT*(K-1))*(IPLAN-1)
01720 100   THR(K)=ANG(IANG)*(IPLAN-1)+(ANG1(IANG)+DT*(K-1))*(2-IPLAN)
01730       WRITE(6,403)
01740 403   FORMAT(1H1,25X,9HPOWER(DB),/,20X,21(1H-),//,
01750      1 39H      PHI     THETA          O             O,/,
01760      2 39H+                            -             /,/)
01770C CALCULATION OF FAR-FIELD RADIATION.
01780       DO 110 K=1,NT
01790       CALL RBOT(RBT,RBP,1,THR(K),PHIR(K))
01800       IC=0
01810       DO 115 MM=1,KMODE
01820       M=-NPODE+MM
01830CC CALCULATE DM AS A FUNCTION OF THETA (EQ. 41).
```

```
C1840         DMT=2.*BL*SINC(M*PI+BKL*COS(THR(K)*DTOR))
C1850         DMZ=DMT-2.*BL*(-1)**M*SINC(BKL*COS(THR(K)*DTOR))
C1860         DO 105 J=1,LS
C1870         IC=IC+1
C1880         DM=DMT
C1890         IF(J.GT.NM) DM=DMZ
C1900         GT(K)=GT(K)+DM*RAT(J)*CR(IC)
C1910  105    GP(K)=GP(K)+DM*RAP(J)*CB(IC)
C1920  115    CONTINUE
C1921         IF(LC.EQ.0) GO TO 106
C1922         CALL RCAP(RCT,RCP,1,THR(K),PHIR(K))
C1923         CALL MULTRC(LC2+LE,RCT,CC,GT(K))
C1924         CALL MULTRC(LC2+LE,RCP,CC,GP(K))
C1925  106    CONTINUE
C1926         IF(NPW.EQ.0) GO TO 107
C1927         CALL RWIRE(RWT,RWP,1,THR(K),PHIR(K))
C1928         CALL MULTRC(LWJ,RWT,CW,GT(K))
C1929         CALL MULTRC(LWJ,RWP,CW,GP(K))
C1930  107    CONTINUE
C1931         X2=CABS(GT(K))
C1940         X2=DBCON(P1*X2*X2)
C1950         X3=CABS(GP(K))
C1960         X3=DBCON(P1*X3*X3)
C1970         WRITE(6,113) PHIR(K),THR(K),X2,X3
C1980  113    FORMAT(1X,2F8.1,2F12.2)
C1990  110    CONTINUE
C2000  90     CONTINUE
C2010C  *** END FAR-FIELD RADIATION ***
C2020C  LIST AND PLOT CURRENTS.
C2050         CALL LPCUR(NMODE,NM,CB,LW,NJ,CW,LC,LE,CC)
C2060C  **** END LISTING AND PLOTTING OF CURRENTS ****
C2080C
C2090C  NEAR FIELD ANALYSIS FOLLOWS.
C2100C
C2110C  NOTE, ESC IS FOR ELECTRIC FIELDS
C2120C         HSC IS FOR MAGNETIC FIELDS
C2130C         INDICES 1,2,AND 3 ARE FOR THE X,Y,AND Z COMPONENTS OF THE
C2140C         FIELDS, RESPECTIVELY.
C2150C
C2160         READ(5,44) NTEST
C2170         IF(NTEST.EQ.0) GO TO 900
C2180         WRITE(6,404)
C2190  404    FORMAT(1H1,//,21H  NEAR FIELD ANALYSIS,//
C2200        1 59X,7HE-FIELD,37X,7HH-FIELD,/,
C2210        2 43X,39(1H-),5X,39(1H-),/,
C2220        3 55X,16HFIELD COMPONENTS,28X,16HFIELD COMPONENTS,//,
C2230        4 24H    ZTEST    YTEST    XTEST,
C2240        5 11X,2(14X,6H   X   ,7X,5H  Y  ,9X,3H Z ),/)
C2250         DO 490 ITEST=1,NTEST
C2260         READ(5,53) ZTEST,YTEST,XTEST
C2262C  UPDATE PLOT ARRAY.
C2264         NPLOT=NPLOT+1
```

```
C2266         XPLOT(NPLOT)=XTEST
C2268         YPLOT(NPLOT)=YTEST
C2270         ZPLOT(NPLOT)=ZTEST
C2272         ISYM(NPLOT)='N'
C2330         ESC(1)=0.
C2340         ESC(2)=0.
C2350         ESC(3)=0.
C2360         HSC(1)=0.
C2370         HSC(2)=0.
C2380         HSC(3)=0.
C2385C  BOT CONTRIBUTION.
C2390C  STEP THROUGH MODES.
C2400         M=-NMODE
C2410         DO 43C MM=1,KMODE
C2420         M=M+1
C2430C  CALCULATE M-TH MODAL CURRENT COEFFICIENTS (EQS. 67 & 71).
C2440         CALL NEARB(XTEST,YTEST,ZTEST,Y)
C2450C  CALCULATE AND SUM Y*CB USING THE FIRST 6 ROWS OF Y, YIELDING THE
C2460C  3 FIELD COMPONENTS FOR BOTH ELECTRIC AND MAGNETIC NEAR FIELDS,
C2470C  (EQS. 67 & 71).
C2480         DO 42C K=1,3
C2490         I1=(P+NMODE-1)*LS
C2500         J1=K-6
C2510         DO 42C I=1,LS
C2520         I1=I1+1
C2530         J1=J1+6
C2540         J2=J1+3
C2545         ESC(K)=ESC(K)+Y(J1)*CB(I1)
C2550         HSC(K)=HSC(K)+Y(J2)*CB(I1)
C2551  420    CONTINUE
C2552  43C    CONTINUE
C2554         IF(NC.EQ.0) GO TO 45C
C2555C  CAP CONTRIBUTION.
C2556         CALL NEARC(XTEST,YTEST,ZTEST,Y)
C2557         DO 44C K=1,3
C2558         J1=K-6
C2559         DO 44C I=1,LC2
C2560         J1=J1+6
C2561         J2=J1+3
C2562         ESC(K)=ESC(K)+Y(J1)*CC(I)
C2563         HSC(K)=HSC(K)+Y(J2)*CC(I)
C2564  44C    CONTINUE
C2565  450    IF(NPW.EQ.0) GO TO 47C
C2566C  WIRE CONTRIBUTION.
C2567         CALL NEARW(XTEST,YTEST,ZTEST,Y)
C2568         DO 46C K=1,3
C2569         J1=K-6
C2570         DO 46C I=1,LW
C2571         J1=J1+6
C2572         J2=J1+3
C2573         ESC(K)=ESC(K)+Y(J1)*CW(I)
C2574         HSC(K)=HSC(K)+Y(J2)*CW(I)
```

```
C2575  460   CONTINUE
C2590  470   WRITE(6,505) ZTEST,YTEST,XTEST
C2600  505   FORMAT(/,1X,3F8.4)
C2780        WRITE(6,506) (CABS(ESC(I)),I=1,3),(CABS(HSC(I)),I=1,3)
C2790  506   FORMAT(43X,3E13.4,5X,3E13.4)
C2800  490   CONTINUE
C2810  900   CONTINUE
C2812        CALL PLOT(XPLOT,YPLOT,ISYM,NPLOT,'X','Y',101,51)
C2814        CALL PLOT(ZPLOT,YPLOT,ISYM,NPLOT,'Z','Y',101,51)
C2816        CALL PLOT(ZPLOT,XPLOT,ISYM,NPLOT,'Z','X',101,51)
C2818        STOP
C2820        END
C2821C -------------------------------------------------------
C2822        SUBROUTINE ZERO(N,A)
C2823C A=C.
C2824        COMPLEX A(1)
C2825        DO 100 I=1,N
C2826        A(I)=0.0
C2827  100   CONTINUE
C2828        RETURN
C2829        END
C2830C -------------------------------------------------------------
C2840        SUBROUTINE MULTYV(NI,NJ,Y,V,C)
C2850C
C2860C UPDATE CURRENT ARRAY C BY Y * V.
C2870C
C2880        COMPLEX S,Y(1),V(1),C(1)
C2890        DO 100 I=1,NI
C2900        S=0.0
C2910        IND=I
C2920        DO 50 J=1,NJ
C2930        S=S+Y(IND)*V(J)
C2940  50    IND=IND+NI
C2950        C(I)=C(I)+S
C2960  100   CONTINUE
C2970        RETURN
C2980        END
C2990C -------------------------------------------------------------
C3000        SUBROUTINE MULTVC(NI,V,C,P)
C3010C
C3020C UPDATE P BY V(CONJUGATE) * C.
C3030C
C3040        COMPLEX V(1),C(1)
C3050        DO 100 I=1,NI
C3060        P=P+CONJG(V(I))*C(I)
C3070  100   CONTINUE
C3080        RETURN
C3090        END
C3100C -------------------------------------------------
C3110        SUBROUTINE MULTRC(NI,R,C,G)
C3120C
C3130C UPDATE G BY R * C.
```

```
C3140C
C3150       COMPLEX G,R(1),C(1)
C3160       DO 100 I=1,NI
C3170       G=G*R(I)*C(I)
C3180 100   CONTINUE
C3190       RETURN
C3200       END
C3210C ************************************************************
C3220       FUNCTION SINC(X)
C3230C ************************************************************
C3240       SINC=1.0
C3250       IF(X.NE.0.0) SINC=SIN(X)/X
C3260       RETURN
C3270       END
C3280C ***************************************************
C3290       FUNCTION DBCON(X)
C3300C ***************************************************
C3310C RETURN X IN DB.
C3320       DBCON=-1000.0
C3330       IF(X.EQ.0.0) RETURN
C3340       DBCON=10.*ALOG10(X)
C3350       RETURN
C3360       END
C3370C -------------------------------------------------------------
C3375       SUBROUTINE BOTIN
C3380C
C3385C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
C3390C
C3392       COMMON /PLOT1/ NPLOT,XPLOT(200),YPLOT(200),ZPLOT(200),ISYM(200)
C3395       COMMON /BOT1/ NMODE,NPT,NBAND
C3400       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C3405       COMMON /BOT3/ DH(82),SV(82),CV(82)
C3410       COMMON /BOT5/ T(160),TP(160),TZ(160)
C3415       COMMON /BOT6/ IEDGE,IUNIT
C3420       READ(5,49) NMODE,NPT,NBAND
C3425       READ(5,49) NP
C3430 49    FORMAT(3I3)
C3435       WRITE(6,48) NMODE,NPT,NBAND,NP
C3440 48    FORMAT(32H     NMODE    NPT    NBAND     NP,/,4I8)
C3445       READ(5,53)(YB(I),I=1,NP)
C3450       READ(5,53)(XB(I),I=1,NP)
C3455   53  FORMAT(10F8.4)
C3460       WRITE(6,55)
C3465 55    FORMAT(/,3H YB)
C3470       WRITE(6,46)(YB(I),I=1,NP)
C3475   46  FORMAT(1X,10F8.4)
C3480       WRITE(6,56)
C3485 56    FORMAT(/,3H XB)
C3486       WRITE(6,46)(XB(I),I=1,NP)
C3488C PLOT THE BODY COORDINATES.
C3490       CALL PLOT8(XB,YB,NP,41)
C3492       READ(5,53) BL
```

```
C3494         WRITE(6,47) BL
C3496  47    FORMAT(//,21H HALF-LENGTH OF BOT =,F12.4)
C3500C UPDATE PLOT ARRAY.
C3501         DO 150 I=1,NP
C3502         NPLOT=NPLOT+1
C3503         XPLOT(NPLOT)=XB(I)
C3504         YPLOT(NPLOT)=YB(I)
C3505         ZPLOT(NPLOT)=BL
C3506         ISYM(NPLOT)='B'
C3507         NPLOT=NPLOT+1
C3508         XPLOT(NPLOT)=XB(I)
C3509         YPLOT(NPLOT)=YB(I)
C3510         ZPLOT(NPLOT)=-BL
C3511         ISYM(NPLOT)='B'
C3512 150    CONTINUE
C3520C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
C3525         IF(MOD(NP,2).NE.1) GO TO 980
C3530C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
C3535         IEDGE=1
C3540         IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
C3545         IEDGE=0
C3550         YB(NP+1)=YB(2)
C3555         XB(NP+1)=XB(2)
C3560         YB(NP+2)=YB(3)
C3565         XB(NP+2)=XB(3)
C3570         NP=NP+2
C3575         WRITE(6,66) NP
C3580  66    FORMAT(//,39H BOT GENERATING CURVE IS CLOSED.  NP = ,I3)
C3585  10    CONTINUE
C3590C COMPUTATION OF BODY SEGMENT PARAMETERS.
C3595         DO 57 I=2,NP
C3600         I2=I-1
C3605         RR1=YB(I)-YB(I2)
C3610         RR2=XB(I)-XB(I2)
C3615         DH(I2)=SQRT(RR1*RR1+RR2*RR2)
C3620         XB1(I2)=.5*(XB(I)+XB(I2))
C3625         YB1(I2)=.5*(YB(I)+YB(I2))
C3630         SV(I2)=RR1/DH(I2)
C3635         CV(I2)=RR2/DH(I2)
C3640  57    CONTINUE
C3645C CHECK IF BOT SEGMENTATION IS UNIFORM.
C3650         IUNIF=0
C3655         NP1=NP-1
C3660         DO 60 I=2,NP1
C3665         RR1=DH(I)/DH(1)
C3670         IF(RR1.LT.0.99 .OR. RR1.GT.1.01) GO TO 20
C3675  60    CONTINUE
C3680         IUNIF=1
C3685         WRITE(6,67)
C3690  67    FORMAT(//,' BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
C3695  20    CONTINUE
C3700C COMPUTATION OF TRIANGLE FUNCTIONS T.
```

```
C3705       NM=(NP-3)/2
C3710       DO 74 J=1,NM
C3715       J2=2*(J-1)+1
C3720       J3=J2+1
C3725       J4=J3+1
C3730       J5=J4+1
C3735       J6=4*(J-1)+1
C3740       J7=J6+1
C3745       J8=J7+1
C3750       J9=J8+1
C3755       DEL1=DH(J2)+DH(J3)
C3760       DEL2=DH(J4)+DH(J5)
C3765       TP(J6)=1./DEL1
C3770       TP(J7)=1./DEL1
C3775       TP(J8)=-1./DEL2
C3780       TP(J9)=-1./DEL2
C3785       T(J6)=DH(J2)/2./DEL1
C3790       T(J7)=(DH(J2)+DH(J3)/2.)/DEL1
C3795       T(J8)=(DH(J4)/2.+DH(J5))/DEL2
C3800       T(J9)=DH(J5)/2./DEL2
C3805    74 CONTINUE
C3810       NM4=NP*4
C3815       DO 75 J=1,NM4
C3820    75 TZ(J)=T(J)
C3825       IF(IEDGE.EQ.0) GO TO 76
C3830       TZ(1)=2.-T(1)
C3835       TZ(2)=2.-T(2)
C3840       TZ(NM4-1)=2.-T(NM4-1)
C3845       TZ(NM4)=2.-T(NM4)
C3850    76 CONTINUE
C3855       RETURN
C3860  980  WRITE(6,981)
C3865  981  FORMAT(//,' **** ERROR IN ROT INPUT')
C3870       STOP
C3875       END
C3880C ***************************************************
C3882       SUBROUTINE PLOTS(X,Y,N,NR)
C3884C ***************************************************************
C3886C
C3888C WRITTEN 2/14/74   BY J. M. PUTNAM        DEPT 220      X23877
C3890C
C3892C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C3894C    N IS THE NUMBER OF POINTS TO BE PLOTTED.
C3896C    NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C3898C    NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C3900C       NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C3902C
C3904       REAL X(1),Y(1),HEAD(10)
C3906       INTEGER  LINE(101),BLANK,STAR,PLUS
C3908       DATA BLANK,STAR,PLUS /1H ,1H*,1H+/
C3910       NC=51
C3912       N10=(NC-1)/10
```

```
C3914          WRITE(6,500)
C3916  500    FORMAT(//,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
C3918          WRITE(6,504)
C3920          XMIN=X(1)
C3922          XMAX=X(1)
C3924          YMIN=Y(1)
C3926          YMAX=Y(1)
C3928          DO 6  I=1,N
C3930          IF(X(I).LT.XMIN) XMIN=X(I)
C3932          IF(X(I).GT.XMAX) XMAX=X(I)
C3934          IF(Y(I).LT.YMIN) YMIN=Y(I)
C3936          IF(Y(I).GT.YMAX) YMAX=Y(I)
C3938  6       CONTINUE
C3940          DEL=XMAX-XMIN
C3942          IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C3944          XMAX=XMIN+DEL
C3946          YMAX=YMIN+DEL
C3948          DO 5  I =1,N10
C3950          Z=I
C3952       5  HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
C3954          DY=(YMAX-YMIN)/(NR-1)
C3956          IEDGE=1
C3958          IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
C3960          Z=YMAX+DY
C3962          YL=Z-DY/2.
C3964          DO 7  J=1,NR
C3966          DO 8  K=1,NC
C3968  8       LINE(K)=BLANK
C3970          Z=Z-DY
C3972          YU=YL
C3974          YL=Z-DY/2.
C3976          DO 9  I=1,N
C3978          IF(Y(I).GE.YU) GO TO 9
C3980          IF(Y(I).LT.YL) GO TO 9
C3982          K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
C3984          IF(K.GT.NC) K=NC
C3986          LINE(K)=STAR
C3988          IF(MOD(I,2).EQ.1) LINE(K)=PLUS
C3990          IF(IEDGE.EQ.0) GO TO 9
C3992          IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
C3994  9       CONTINUE
C3996          WRITE(6,508) Z,(LINE(K),K=1,NC)
C3998  7       CONTINUE
C4000          WRITE(6,504)
C4002          WRITE( 6,3002)
C4004          WRITE( 6,507) XMIN,(HEAD(I),I=1,N10)
C4006          RETURN
C4008   504  FORMAT ( 1X, 14(1H-), 1H+, 10(5H----+), 1H- )
C4010   507  FORMAT(10X,11(F10.4))
C4012   508  FORMAT (1X, F12.4,1X, 1HI, 51A1, 1HI )
C4014  3002  FORMAT(4X,7HYH / XH,4X,1HI,5(9X,1HI))
C4016          END
```

```
C4410C ------------------------------------------------------
C4420       SUBROUTINE CAPIN
C4430C
C4440C READ CAP INPUTS AND COMPUTE CAP ARRAYS.
C4450C
C4460       COMMON /PLOT1/ NPLOT,XPLOT(200),YPLOT(200),ZPLOT(200),ISYM(200)
C4470       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C4480       COMMON /CAP1/ NC,XC,YC,ZC(2)
C4490       COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
C4500       COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
C4510       COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
C4520       COMMON /EDG1/ NE,ZE(2),ZBE(10)
C4530       COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
C4540       READ(5,1) NC,NPR,NF
C4550  1    FORMAT(3I3)
C4560       IF(NE.NE.0) NE=NC
C4570       WRITE(6,3) NC,NPR,NE
C4580  3    FORMAT(16H1   NC   NPR    NE,/,3I5,//)
C4590       IF(NC.EQ.0) RETURN
C4600       READ(5,2) XC,YC
C4610  2    FORMAT(10F8.4)
C4620       READ(5,2) (ZC(I),I=1,NC)
C4630       READ(5,2) (RHOC(I),I=1,NPR)
C4640       IF(NE.NE.0) READ(5,2) (ZE(I),I=1,NE)
C5100C UPDATE PLOT ARRAY.
C5102       DO 150 I=1,NC
C5104       NPLOT=NPLOT+1
C5106       XPLOT(NPLOT)=XC
C5108       YPLOT(NPLOT)=YC
C5110       ZPLOT(NPLOT)=ZC(I)
C5112       ISYM(NPLOT)='C'
C5114       IF(NE.EQ.0) GO TO 150
C5116       NPLOT=NPLOT+1
C5118       XPLOT(NPLOT)=XB(1)
C5120       YPLOT(NPLOT)=YB(1)
C5122       ZPLOT(NPLOT)=ZE(I)
C5124       ISYM(NPLOT)='E'
C5126 150   CONTINUE
C5144       WRITE(6,4)
C5146  4    FORMAT(37H  CAP       XC       YC       ZC       ZE,/)
C5148       DO 160 I=1,NC
C5149       IF(NE.EQ.0) WRITE(6,5) I,XC,YC,ZC(I)
C5150       IF(NE.NE.0) WRITE(6,5) I,XC,YC,ZC(I),ZE(I)
C5151 160   CONTINUE
C5152  5    FORMAT(I4,4X,4F8.4)
C5154       WRITE(6,6)
C5156  6    FORMAT(//,5H RHOC)
C5158       WRITE(6,7) (RHOC(I),I=1,NPR)
C5160  7    FORMAT(1X,10F8.4)
C5162       IF(MCO(NPR,2).NE.1) GO TO 980
C5164       DO 120 I=2,NPR
C5166 120   IF(RHOC(I).LE.RHOC(I-1)) GO TO 980
```

```
C5168C COMPUTATION OF CAP SECTOR PARAMETERS.
C5170        DO 47 I=1,NP
C5172        RC(I)=SQRT((YB(I)-YC)**2+(XB(I)-XC)**2)
C5174 47     CONTINUE
C5176        DO 57 I=2,NP
C5178        I2=I-1
C5180        RR1=YB1(I2)-YC
C5182        RR2=XB1(I2)-XC
C5184        RC1(I2)=SQRT(RR1*RR1+RR2*RR2)
C5186        AC(I2)=ABS((XB(I)-XC)*(YB(I)+YC)+(XB(I2)-XB(I))*
C5188       1 (YB(I2)+YB(I))+(XC-XB(I2))*(YC+YB(I2)))/2.
C5190        SPC(I2)=RR1/RC1(I2)
C5192        CPC(I2)=RR2/RC1(I2)
C5194 57     CONTINUE
C5196        DO 67 I=2,NPR
C5198        I2=I-1
C5200        RHOC1(I2)=(RHOC(I)+RHOC(I2))/2.
C5202        DRHOC(I2)=RHOC(I)-RHOC(I2)
C5204 67     CONTINUE
C5206C COMPUTATION OF CAP TRIANGLE FUNCTIONS.
C5208        LC=(NPR-3)/2
C5210        DO 74 J=1,LC
C5212        J2=2*(J-1)+1
C5214        J3=J2+1
C5216        J4=J3+1
C5218        J5=J4+1
C5220        J6=4*(J-1)+1
C5222        J7=J6+1
C5224        J8=J7+1
C5226        J9=J8+1
C5228        DEL1=DRHOC(J2)+DRHOC(J3)
C5230        DEL2=DRHOC(J4)+DRHOC(J5)
C5232        TPCR(J6)=1./DEL1
C5234        TPCR(J7)=1./DEL1
C5236        TPCR(J8)=-1./DEL2
C5238        TPCR(J9)=-1./DEL2
C5240        TCR(J6)=DRHOC(J2)/2./DEL1
C5242        TCR(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
C5244        TCR(J8)=(DRHOC(J4)/2.+DRHOC(J5))/DEL2
C5246        TCR(J9)=DRHOC(J5)/2./DEL2
C5248   74   CONTINUE
C5250        LC4=LC*4
C5252        DO 75 I=1,LC4
C5254        TCT(I)=TCR(I)
C5256        TPCT(I)=TPCR(I)
C5258 75     CONTINUE
C5260        TCT(LC4-1)=2.-TCT(LC4-1)
C5262        TCT(LC4)=2.-TCT(LC4)
C5264        TPCT(LC4-1)=-TPCT(LC4-1)
C5266        TPCT(LC4)=-TPCT(LC4)
C5268        IF(RHOC(1).EQ.0.0) GO TO 76
C5270        TCT(1)=2.-TCT(1)
```

```
C5272        TCT(2)=2.-TCT(2)
C5274        TPCT(1)=-TPCT(1)
C5276        TPCT(2)=-TPCT(2)
C5278  76    CONTINUE
C5279        IF(NE.EQ.0) RETURN
C5280C COMPUTATION OF EDGE HALF TRIANGLE FUNCTIONS.
C5282        DO 80 IC=1,NC
C5284        J2=NPR-2
C5286        J3=J2+1
C5288        J6=2*(IC-1)+1
C5290        J7=J6+1
C5292        DEL1=DRHOC(J2)+DRHOC(J3)
C5294        TPCE(J6)=1./DEL1
C5296        TPCE(J7)=1./DEL1
C5298        TCE(J6)=DRHOC(J2)/2./DEL1
C5300        TCE(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
C5302        DEL2=ZE(IC)-ZC(IC)
C5304        ZBE(J6)=ZC(IC)+0.25*DEL2
C5306        ZBE(J7)=ZC(IC)+0.75*DEL2
C5308        IF(DEL2.LT.0.0) GO TO 78
C5310C EDGE IS AT Z=-L.
C5312        TPBE(J6)=-1./DEL2
C5314        TPBE(J7)=-1./DEL2
C5316        TBE(J6)=0.75
C5318        TBE(J7)=0.25
C5320        GO TO 80
C5322  78    CONTINUE
C5324C EDGE IS AT Z=+L.
C5326        TPBE(J6)=1./DEL2
C5328        TPBE(J7)=1./DEL2
C5330        TBE(J6)=-0.75
C5332        TBE(J7)=-0.25
C5334  80    CONTINUE
C5336        RETURN
C5338  980   WRITE(6,981)
C5340  981   FORMAT(//,' **** ERROR IN CAP INPUT')
C5342        STOP
C5344        END
C5510C -------------------------------------------------
C5515        SUBROUTINE WIREIN
C5520C
C5525C READ WIRE COORDINATES AND COMPUTE WIRE SEGMENT ARRAYS.
C5530C
C5532        COMMON /PLOT1/ NPLOT,XPLOT(200),YPLOT(200),ZPLOT(200),ISYM(200)
C5535        COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
C5540       1  XW1(100),YW1(100),ZW1(100)
C5545        COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
C5550        COMMON /WIRE3/ NW,INDW(6),RADW(5)
C5555        COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
C5560        COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
C5565        COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
C5570        COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
```

```
      COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
      NW=0
      NJ=0
      LW=0
      READ(5,51) NW,NPW,NJ
   51 FORMAT(3I3)
      WRITE(6,1) NW,NPW,NJ
    1 FORMAT(24H1      NW      NPW      NJ,/,3I8)
      IF(NW.EQ.0) RETURN
      READ(5,53) (XW(I),I=1,NPW)
      READ(5,53) (YW(I),I=1,NPW)
      READ(5,53) (ZW(I),I=1,NPW)
   53 FORMAT(10F8.4)
      READ(5,52) (INDW(I),I=1,NW)
   52 FORMAT(10I8)
      INDW(NW+1)=NPW+1
      READ(5,53) (RADW(I),I=1,NW)
      IF(NJ.EQ.0) GO TO 50
C     NOTE, INDJW MUST BE MONOTONIC INCREASING.
      READ(5,52) (INDJW(I),I=1,NJ)
      READ(5,53) (RADJ(I),I=1,NJ)
      READ(5,53) (UXJ(I),I=1,NJ)
      READ(5,53) (UYJ(I),I=1,NJ)
      READ(5,53) (UZJ(I),I=1,NJ)
   50 IERW=0
      IERJ=0
      WRITE(6,61)
   61 FORMAT(/,22X,'WIRE COORDINATES',20X,'JUNCTION PARAMETERS',/,
     1 26X,'IW',4X,'XW',6X,'YW',6X,'ZW',
     2 10X,'IJ',4X,'RADJ',5X,'UXJ',5X,'UYJ',5X,'UZJ')
      IJ=1
C     THIS LOOP LISTS WIRE/JUNCTION POINTS, WHILE CHECKING THE FOLLOWING:
C       1) EACH WIRE MUST CONTAIN AN ODD NUMBER OF POINTS.
C       2) EACH JUNCTION MUST EITHER START OR TERMINATE A WIRE.
C       3) CHECK THAT ALL JUNCTION POINTS ARE FOUND.
      DO 100 IW=1,NW
      WRITE(6,62) IW,RADW(IW)
   62 FORMAT(2X,'WIRE',I3,' RADW=',F8.4)
      I1=INDW(IW)
      I2=INDW(IW+1)-1
C     CHECK FOR AN ODD NUMBER OF POINTS ON WIRE IW.
      IF(MOD(I2-I1+1,2).NE.1) IERW=1
      DO 90 I=I1,I2
      WRITE(6,63) I,XW(I),YW(I),ZW(I)
   63 FORMAT(25X,I3,3F8.4)
C     UPDATE PLOT ARRAY.
      NPLOT=NPLOT+1
      XPLOT(NPLOT)=XW(I)
      YPLOT(NPLOT)=YW(I)
      ZPLOT(NPLOT)=ZW(I)
```

```
C5811         ISYM(NPLOT)='W'
C5814C CHECK IF WIRE POINT I IS A JUNCTION POINT.
C5815         IF(IJ.GT.NJ) GO TO 90
C5820         IF(INDJW(IJ).NE.I) GO TO 90
C5825C CHECK THAT JUNCTION POINT IJ IS AT THE START OR END OF WIRE.
C5830         IF(I.NE.I1 .AND. I.NE.I2) IERJ=1
C5835         WRITE(6,64) IJ,RADJ(IJ),UXJ(IJ),UYJ(IJ),UZJ(IJ)
C5840 64    FORMAT('+',58X,I3,4F8.4)
C5842C UPDATE PLOT ARRAY.
C5843         ISYM(NPLOT)='J'
C5845C DETERMINE DIRECTION IN WHICH WIRE LEAVES JUNCTION POINT.
C5850         INDTJ(IJ)=1
C5855         IF(I.EQ.I2) INDTJ(IJ)=-1
C5860C COMPUTE JUNCTION PARAMETERS.
C5865         RADJ(IJ)=RADW(IW)
C5870         XJ(IJ)=XW(I)
C5875         YJ(IJ)=YW(I)
C5880         ZJ(IJ)=ZW(I)
C5885         IJ=IJ+1
C5890 90    CONTINUE
C5895 100   CONTINUE
C5900C CHECK FOR WIRE OR JUNCTION INPUT ERRORS.
C5905         IF(IERW.NE.0) GO TO 980
C5910         IF(IJ-1.NE.NJ .OR. IERJ.NE.0) GO TO 990
C5915C COMPUTATION OF WIRE SEGMENT PARAMETERS.
C5920         DO 57 I=2,NPW
C5925         I2=I-1
C5930         DXW(I2)=XW(I)-XW(I2)
C5935         DYW(I2)=YW(I)-YW(I2)
C5940         DZW(I2)=ZW(I)-ZW(I2)
C5945         DHW(I2)=SQRT(DXW(I2)**2+DYW(I2)**2+DZW(I2)**2)
C5950         XW1(I2)=0.5*(XW(I)+XW(I2))
C5955         YW1(I2)=0.5*(YW(I)+YW(I2))
C5960         ZW1(I2)=0.5*(ZW(I)+ZW(I2))
C5965      57 CONTINUE
C5970C COMPUTATION OF WIRE TRIANGLE FUNCTIONS TW.
C5975         LW=0
C5980         DO 75 IW=1,NW
C5965         I1=INDW(IW)
C5990         I2=INDW(IW+1)-1
C5995         LW1=(I2-I1-2)/2
C6000         DO 74 J=1,LW1
C6005         LW=LW+1
C6010         J2=2*(J-1)+I1
C6015         J3=J2+1
C6020         J4=J3+1
C6025         J5=J4+1
C6030         J6=4*(LW-1)+1
C6035         J7=J6+1
C6040         J8=J7+1
C6045         J9=J8+1
C6050         INDTW(LW)=J2
```

```
C6055       DEL1=DHW(J2)+DHW(J3)
C6060       DEL2=DHW(J4)+DHW(J5)
C6065       TPW(J6)=1./DEL1
C6070       TPW(J7)=1./DEL1
C6075       TPW(J8)=-1./DEL2
C6080       TPW(J9)=-1./DEL2
C6085       TW(J6)=DHW(J2)/2./DEL1
C6090       TW(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C6095       TW(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C6100       TW(J9)=DHW(J5)/2./DEL2
C6105 74    CONTINUE
C6110 75    CONTINUE
C6115C NOTE, LW=(NPW-NW)/2-NW
C6120       IF(NJ.EQ.0) RETURN
C6125C COMPUTATION OF JUNCTION HALF TRIANGLE FUNCTIONS TJ.
C6130       DO 85 IJ=1,NJ
C6135       IF(INDTJ(IJ).GT.0) GO TO 80
C6140C JUNCTION IS AT THE END OF A WIRE.
C6145       J2=INDJW(IJ)-2
C6150       J3=J2+1
C6155       J6=2*(IJ-1)+1
C6160       J7=J6+1
C6165       INDTJ(IJ)=J2
C6170       DEL1=-DHW(J2)-DHW(J3)
C6175       TPJ(J6)=1./DEL1
C6180       TPJ(J7)=1./DEL1
C6185       TJ(J6)=DHW(J2)/2./DEL1
C6190       TJ(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C6195       GO TO 85
C6200C JUNCTION IS AT THE START OF A WIRE.
C6205 80    J4=INDJW(IJ)
C6210       J5=J4+1
C6215       J8=2*(IJ-1)+1
C6220       J9=J8+1
C6225       INDTJ(IJ)=J4
C6230       DEL2=DHW(J4)+DHW(J5)
C6235       TPJ(J8)=-1./DEL2
C6240       TPJ(J9)=-1./DEL2
C6245       TJ(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C6250       TJ(J9)=DHW(J5)/2./DEL2
C6255 85    CONTINUE
C6260C COMPUTE UNIT VECTORS FOR JUNCTION(DISK).
C6265       DO 95 IJ=1,NJ
C6270C NORMAL UNIT VECTOR (UJ).
C6275       RR=SQRT(UXJ(IJ)**2+UYJ(IJ)**2+UZJ(IJ)**2)
C6280       UXJ(IJ)=UXJ(IJ)/RR
C6285       UYJ(IJ)=UYJ(IJ)/RR
C6290       UZJ(IJ)=UZJ(IJ)/RR
C6295C FIND 2 ORTHOGONAL UNIT VECTORS IN THE PLANE OF THE DISK (UJ1 & UJ2).
C6300       UXJ1(IJ)=0.0
C6305       UYJ1(IJ)=0.0
C6310       UZJ1(IJ)=0.0
```

```
      UXJ2(IJ)=0.0
      UYJ2(IJ)=0.0
      UZJ2(IJ)=0.0
      IF(UXJ(IJ).EQ.0.0) GO TO 91
C FIND INTERSECTION WITH X-Y PLANE.
      UYJ1(IJ)=1.0
      UXJ1(IJ)=-UYJ(IJ)/UXJ(IJ)
C FIND INTERSECTION WITH X-Z PLANE.
      UZJ2(IJ)=1.0
      UXJ2(IJ)=-UZJ(IJ)/UXJ(IJ)
      GO TO 94
91    IF(UYJ(IJ).EQ.0.0) GO TO 92
C FIND INTERSECTION WITH Y-Z PLANE.
      UZJ1(IJ)=1.0
      UYJ1(IJ)=-UZJ(IJ)/UYJ(IJ)
C FIND INTERSECTION WITH X-Y PLANE.
      UXJ2(IJ)=1.0
      UYJ2(IJ)=-UXJ(IJ)/UYJ(IJ)
      GO TO 94
92    IF(UZJ(IJ).EQ.0.0) GO TO 94
C FIND INTERSECTION WITH X-Z PLANE.
      UXJ1(IJ)=1.0
      UZJ1(IJ)=-UXJ(IJ)/UZJ(IJ)
C FIND INTERSECTION WITH Y-Z PLANE.
      UYJ2(IJ)=1.0
      UZJ2(IJ)=-UYJ(IJ)/UZJ(IJ)
94    CONTINUE
      RR=SQRT(UXJ1(IJ)**2+UYJ1(IJ)**2+UZJ1(IJ)**2)
      UXJ1(IJ)=UXJ1(IJ)/RR
      UYJ1(IJ)=UYJ1(IJ)/RR
      UZJ1(IJ)=UZJ1(IJ)/RR
      RR=SQRT(UXJ2(IJ)**2+UYJ2(IJ)**2+UZJ2(IJ)**2)
      UXJ2(IJ)=UXJ2(IJ)/RR
      UYJ2(IJ)=UYJ2(IJ)/RR
      UZJ2(IJ)=UZJ2(IJ)/RR
95    CONTINUE
      RETURN
980   WRITE(6,981)
981   FORMAT(//,' **** ERROR IN WIRE INPUT')
      STOP
990   WRITE(6,991)
991   FORMAT(//,' **** ERROR IN JUNCTION INPUT')
      STOP
      END
C ------------------------------------------------------------
      SUBROUTINE VWIRE(NWJV,VW)
C
C READ WIRE AND JUNCTION VOLTAGES.
C
      DIMENSION IW(10)
      COMPLEX EW(10)
      COMPLEX VW(1)
```

```
      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
      K=LW+NJ
      DO 25 I=1,K
25    VW(I)=0.0
      WRITE(6,1)
1     FORMAT(1H1)
      READ(5,2) NWJV
2     FORMAT(I3)
      IF(NWJV.EQ.0) RETURN
      READ(5,3) (IW(I),I=1,NWJV)
3     FORMAT(10I8)
      READ(5,4) (EW(I),I=1,NWJV)
4     FORMAT(10F8.4)
      WRITE(6,5)
5     FORMAT('     WIRE VOLTAGES',//,'  WIRE INDEX',13X,'VOLTAGE')
      DO 50 I=1,NWJV
      WRITE(6,6) IW(I),EW(I)
6     FORMAT(1X,I6,9X,2F10.4)
      VW(IW(I))=EW(I)
50    CONTINUE
      RETURN
      END
C-----------------------------------------------------------
      SUBROUTINE SLOTIN
C
C READ SLOT ANTENNA INPUTS.
C
      COMPLEX EO
      COMMON /PLOT1/ NPLOT,XPLOT(200),YPLOT(200),ZPLOT(200),ISYM(200)
      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
      COMMON /SLOT1/ NSA,IS(20),ZO(20),Z1(20)
      COMMON /SLOT2/ EO(20),TEXC(20),ZEXC(20)
      READ(5,16) NSA
16    FORMAT(I3)
      WRITE(6,3) NSA
3     FORMAT(//,26H1NUMBER OF SLOT ANTENNAS =,I3)
      IF(NSA.EQ.0) RETURN
      READ(5,54) (IS(K),K=1,NSA)
54    FORMAT(10I8)
C NOTE -L <= ZO(K) < Z1(K) <= L.
      READ(5,53) (ZO(K),K=1,NSA)
53    FORMAT(10F8.4)
      READ(5,53) (Z1(K),K=1,NSA)
      READ(5,53) (EO(K),K=1,NSA)
C T-EXCITATION.
      READ(5,53) (TEXC(K),K=1,NSA)
C Z-EXCITATION.
      READ(5,53) (ZEXC(K),K=1,NSA)
      WRITE(6,4) ((K,IS(K),ZO(K),Z1(K),EO(K),TEXC(K),ZEXC(K)),K=1,NSA)
4     FORMAT(//,50H ANTENNA NO.    IS      ZO          Z1                    EO,
     1 11X,14HTEXC       ZEXC,//,(I9,I8,6F10.4))
```

```
C7911C UPDATE PLOT ARRAY.
C7912        DO 150 I=1,NSA
C7913        NPLOT=NPLOT+1
C7914        IND=IS(I)
C7915        XPLOT(NPLOT)=XB(2*IND+1)
C7916        YPLOT(NPLOT)=YB(2*IND+1)
C7917        ZPLOT(NPLOT)=ZO(I)
C7918        ISYM(NPLOT)='S'
C7919        NPLOT=NPLOT+1
C7920        XPLOT(NPLOT)=XB(2*IND+1)
C7921        YPLOT(NPLOT)=YB(2*IND+1)
C7922        ZPLOT(NPLOT)=Z1(I)
C7923        ISYM(NPLOT)='S'
C7924 150    CONTINUE
C7925        RETURN
C7930        END
C7940C -------------------------------------------------------
C7950        SUBROUTINE ORDER(MORD)
C7960C DETERMINE THE ORDER IN WHICH THE Y MATRIX WAS GENERATED, AND
C7970C CHECK FOR DIMENSIONAL CONSISTENCY BETWEEN THE Y MATRIX DATA FILE,
C7980C AND THE INPUT DATA FILE.
C7990        DIMENSION MORD(2)
C8000        COMMON /BOT1/ NMODE,NPT,NBAND
C8010        COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C8020        COMMON /CAP1/ NC,XC,YC,ZC(2)
C8025        COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
C8030        COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
C8032       1 XW1(100),YW1(100),ZW1(100)
C8034        COMMON /WIRE3/ NW,INDW(6),RADW(5)
C8040        COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
C8042        COMMON /EDG1/ NE,ZE(2),ZBE(10)
C8050        READ(1) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
C8052        IF(NMODE.NE.NMODE1) GO TO 500
C8054        IF(NP.NE.NP1) GO TO 500
C8056        IF(NC.NE.NC1) GO TO 500
C8058        IF(NPR.NE.NPR1) GO TO 500
C8060        IF(NE.NE.NE1) GO TO 500
C8062        IF(NW.NE.NW1) GO TO 500
C8064        IF(NPW.NE.NPW1) GO TO 500
C8066        IF(NJ.NE.NJ1) GO TO 500
C8120C MORD SPECIFIES THE ORDER IN WHICH CAPS AND/OR WIRES HAVE BEEN ADDED
C8130C TO THE BOT.  INDEX 1 REFERS TO CAPS AND INDEX 2 REFERS TO WIRES.
C8140C   MORD(I)=0 IF I HAS NOT BEEN ADDED
C8150C   MORD(I)=N IF I HAS BEEN ADDED TO THE SYSTEM MATRIX, AND
C8160C   IS LOCATED AT PSEUDO MODE NUMBER N.
C8170        READ(1) MORD
C8220        WRITE(6,4) MORD
C8230 4      FORMAT(' THE Y MATRIX CONTAINS THE FOLLOWING ADDITIONS ( 0 IF
C8240       1 ' NOT INCLUDED, OR CORRESPONDING MODE NUMBER IF PRESENT):',///,
C8250       2 10X,'CAPS',I4,/,10X,'WIRES',I3)
C8260        RETURN
C8270 500    WRITE(6,1)
```

```
 1     FORMAT(//,' *** ERROR ***   INPUT PARAMETERS DO NOT CHECK WITH',
     1 ' THE PARAMETERS IN THE Y MATRIX FILE',//,
     2 6X,'  NMODE   NP    NC    NPR    NE    NW   NPW    NJ')
       WRITE(6,2) NMODE,NP,NC,NPR,NE,NW,NPW,NJ
 2     FORMAT(' INPUT ',8I6)
       WRITE(6,3) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
 3     FORMAT(' Y FILE',8I6)
       STOP
       END
C-----------------------------------------------------------------
       SUBROUTINE VBOT(M,VB)
C
C COMPUTE BOT VOLTAGE FOR MODE M.
C
       COMPLEX U,A3,A4,VB(1)
       COMPLEX EO
       COMMON /SLOT1/ NSA,IS(20),ZO(20),Z1(20)
       COMMON /SLOT2/ EO(20),TEXC(20),ZEXC(20)
       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
       COMMON /BOT3/ DH(82),SV(82),CV(82)
       DATA U,PI /(0.,1.),3.14159265/
       LS=NP-3
       NM=LS/2
       DO 50 I=1,LS
 50    VB(I)=0.0
       DO 70 K=1,NSA
       I=IS(K)
       I1=I+NM
       J=2*(I-1)
       DH4=DH(J+1)+DH(J+2)+DH(J+3)+DH(J+4)
       A3=Z1(K)-ZO(K)
       IF(M.EQ.0) GO TO 66
       A3=-U*BL/M/PI*(CEXP(U*M*PI*Z1(K)/BL)-CEXP(U*M*PI*ZO(K)/BL))
 66    A4=A3-(-1)**M*(Z1(K)-ZO(K))
       VB(I)=VB(I)+EO(K)*DH4/2.*A3*TEXC(K)
       VB(I1)=VB(I1)+EO(K)*DH4/2.*A4*ZEXC(K)
 70    CONTINUE
       RETURN
       END
C*****************************************************************
       SUBROUTINE RBOT(RBT,RBP,NT,TH,PHI)
C*****************************************************************
C COMPUTE BOT TRANSFER MATRICES FOR THETA AND PHI POLARIZATION.
       COMPLEX RBT(1),RBP(1),A6,U
       COMMON /WAVE/ BK
       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
       COMMON /BOT3/ D4(82),SV(82),CV(82)
       COMMON /BOT5/ T(160),TP(160),TZ(160)
       DIMENSION TH(1),PHI(1)
       DATA PI,U /3.14159265,(0.,1.)/
       DATA DTOR /0.017453292/
       LS=NP-3
```

```
C8800       NM=LS/2
C8810       DO 156 K=1,NT
C8820       K1=(K-1)*LS
C8830       CS=COS(TH(K)*DTOR)
C8840       SN=SIN(TH(K)*DTOR)
C8850       CSPHI=COS(PHI(K)*DTOR)
C8860       SNPHI=SIN(PHI(K)*DTOR)
C8870       DO 300 J=1,NM
C8880       J1=J+K1
C8890       J2=J1+NM
C8900       RBT(J1)=0.
C8910       RBT(J2)=0.
C8920       RBP(J1)=0.
C8930       RBP(J2)=0.
C8940C CALCULATION OF TRANSFER MATRICES (EQS. 38-41).
C8950C NOTE, ONLY THE MODE INDEPENDENT PORTION IS COMPUTED HERE.
C8960       DO 301 I=1,4
C8970       I1=2*(J-1)+I
C8980       I4=4*(J-1)+I
C8990       A6=CEXP(U*BK*SN*(XB1(I1)*CSPHI+YB1(I1)*SNPHI))
C9000C THETA POLARIZED (EQS. 38-39).
C9010       RBT(J1)=RBT(J1)+DH(I1)*T(I4)*(CV(I1)*CSPHI+SV(I1)*SNPHI)*A6
C9020       RBT(J2)=RBT(J2)+DH(I1)*TZ(I4)*A6
C9030C PHI POLARIZED (EQS. 40-41).
C9040       RBP(J1)=RBP(J1)+DH(I1)*T(I4)*(SV(I1)*CSPHI-CV(I1)*SNPHI)*A6
C9050   301 CONTINUE
C9060       RBT(J1)=CS*RBT(J1)
C9070       RBT(J2)=-SN*RBT(J2)
C9080   300 CONTINUE
C9090   156 CONTINUE
C9100       RETURN
C9110       END
C9111C ------------------------------------------------------
C9112       SUBROUTINE RCAP(RCT,RCP,NT,TH,PHI)
C9113C
C9114C COMPUTE CAP TRANSFER MATRICES FOR THETA AND PHI POLARIZATION.
C9115C
C9116       COMPLEX RCT(1),RCP(1),A6,U
C9117       COMMON /WAVE/ BK
C9118       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C9119       COMMON /BOT3/ DH(82),SV(82),CV(82)
C9120       COMMON /BOT5/ T(160),TP(160),TZ(160)
C9121       COMMON /CAP1/ NC,XC,YC,ZC(2)
C9122       COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
C9123       COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
C9124       COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
C9125       COMMON /EDG1/ NE,ZE(2),ZBE(10)
C9126       COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
C9127       DIMENSION TH(1),PHI(1)
C9128       DATA PI,U /3.14159265,(0.,1.)/
C9129       DATA DTOR /0.0174532927
C9130       NM=(NP-3)/2
```

```
C9131        LR=(NPR-3)/2
C9132        LC=NC*NM*LR
C9133        LE=NE*NM
C9134        DO 156 K=1,NT
C9135        K1=(K-1)*(2*LC+LE)
C9136        CS=COS(TH(K)*DTOR)
C9137        SN=SIN(TH(K)*DTOR)
C9138        CSPHI=COS(PHI(K)*DTOR)
C9139        SNPHI=SIN(PHI(K)*DTOR)
C9140        J1=K1
C9141        DO 300 JC=1,NC
C9142        DO 300 JP=1,NM
C9143        DO 300 JR=1,LR
C9144        J1=J1+1
C9145        J2=J1+LC
C9146        RCT(J1)=0.
C9147        RCT(J2)=0.
C9148        RCP(J1)=0.
C9149        RCP(J2)=0.
C9150C CALCULATION OF TRANSFER MATRICES.
C9151        DO 301 IIP=1,4
C9152        IP1=2*(JP-1)+IIP
C9153        IP4=4*(JP-1)+IIP
C9154        DO 301 IIR=1,4
C9155        IR1=2*(JR-1)+IIR
C9156        IR4=4*(JR-1)+IIR
C9157        XX=XC+RHOC1(IR1)*RC1(IP1)*CPC(IP1)
C9158        YY=YC+RHOC1(IR1)*RC1(IP1)*SPC(IP1)
C9159        ZZ=ZC(JC)
C9160        A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
C9161        AI=AC(IP1)*ABS(RHOC(IR1+1)**2-RHOC(IR1)**2)
C9162C THETA POLARIZED.
C9163        RCT(J1)=RCT(J1)+AI*(CV(IP1)*CSPHI+SV(IP1)*SNPHI)*
C9164       1 T(IP4)*TCT(IR4)*A6
C9165        RCT(J2)=RCT(J2)+AI*(CPC(IP1)*CSPHI+SPC(IP1)*SNPHI)*
C9166       1 TZ(IP4)*TCR(IR4)*A6
C9167C PHI POLARIZED.
C9168        RCP(J1)=RCP(J1)+AI*(SV(IP1)*CSPHI-CV(IP1)*SNPHI)*
C9169       1 T(IP4)*TCT(IR4)*A6
C9170        RCP(J2)=RCP(J2)+AI*(SPC(IP1)*CSPHI-CPC(IP1)*SNPHI)*
C9171       1 TZ(IP4)*TCR(IR4)*A6
C9172  301   CONTINUE
C9173        RCT(J1)=CS*RCT(J1)
C9174        RCT(J2)=CS*RCT(J2)
C9175  300   CONTINUE
C9176        IF(NE.EQ.0) GO TO 156
C9177C EDGE TRANSFER MATRICES FOLLOW.
C9178        J2=2*LC+K1
C9179        DO 400 JC=1,NC
C9180        DO 400 JP=1,NM
C9181        J2=J2+1
C9182        RCT(J2)=0.
```

```
09183       RCP(JZ)=0.
09184       DO 401 IIP=1,4
09185       IP1=2*(JP-1)+IIP
09186       IP4=4*(JP-1)+IIP
09187       DO 401 IIRZ=1,2
09188       IR1=2*LR+IIRZ
09189       IR4=2*(JC-1)+IIRZ
09190       IZ1=2*(JC-1)+IIRZ
09191       IZ4=2*(JC-1)+IIRZ
09192       XX=XC+RHOC1(IR1)*RC1(IP1)*CPC(IP1)
09193       YY=YC+RHOC1(IR1)*RC1(IP1)*SPC(IP1)
09194       ZZ=ZC(JC)
09195       A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
09196       AI=AC(IP1)*ABS(RHOC(IR1+1)**2-RHOC(IR1)**2)
09197C EDGE(CAP)
09198C THETA POLARIZED.
09199       RCT(JZ)=RCT(JZ)+CS*AI*(CPC(IP1)*CSPHI+SPC(IP1)*SNPHI)*
09200      1 TZ(IP4)*TCE(IR4)*A6
09201C PHI  POLARIZED.
09202       RCP(JZ)=RCP(JZ)+AI*(SPC(IP1)*CSPHI-CPC(IP1)*SNPHI)*
09203      1 TZ(IP4)*TCE(IR4)*A6
09204       XX=XB1(IP1)
09205       YY=YB1(IP1)
09206       ZZ=ZBE(IZ1)
09207       A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
09208       AI=DH(IP1)*ABS(ZE(JC)-ZC(JC))/2.
09209C EDGE(BOT).
09210C THETA POLARIZED.
09211       RCT(JZ)=RCT(JZ)-SN*AI*TZ(IP4)*TBE(IZ4)*A6
09212  4C1  CONTINUE
09213  400  CONTINUE
09214  156  CONTINUE
09215       RETURN
09216       END
10170C -------------------------------------------------
10180       SUBROUTINE RWIRE(RWT,RWP,NT,TH,PHI)
10190C
10200C COMPUTE WIRE-JUNCTION TRANSFER MATRICES.
10210C
10220       COMPLEX RWT(1),RWP(1),A6,U
10230       COMMON /WAVE/ BK
10240       COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
10250      1 XW1(100),YW1(100),ZW1(100)
10260       COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
10270       COMMON /WIRE3/ NW,INDW(6),RADW(5)
10280       COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
10290       COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
10300       COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
10310       COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
10320       COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
10330       COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
10340       DIMENSION TH(1),PHI(1)
```

```
      DATA PI,U /3.14159265,(0.,1.)/
      DATA DTOR /0.017453292/
      LWJ=LW+NJ
      DO 156 K=1,NT
      K1=(K-1)*LWJ
      CS=COS(TH(K)*DTOR)
      SN=SIN(TH(K)*DTOR)
      CSPHI=COS(PHI(K)*DTOR)
      SNPHI=SIN(PHI(K)*DTOR)
      DO 300 J=1,LW
      J1=J+K1
      RWT(J1)=0.
      RWP(J1)=0.
      DO 301 I=1,4
      I1=INDTW(J)+I-1
      I4=4*(J-1)+I
      A6=CEXP(U*BK*(SN*(XW1(I1)*CSPHI+YW1(I1)*SNPHI)+ZW1(I1)*CS))
      RWT(J1)=RWT(J1)+(CS*(DXW(I1)*CSPHI+DYW(I1)*SNPHI)-
     1 DZW(I1)*SN)*TW(I4)*A6
      RWP(J1)=RWP(J1)+(-DXW(I1)*SNPHI+DYW(I1)*CSPHI)*TW(I4)*A6
  301 CONTINUE
  300 CONTINUE
      IF(NJ.EQ.0) GO TO 156
      J1=LW+K1
      DO 400 J=1,NJ
      J1=J1+1
      RWT(J1)=0.
      RWP(J1)=0.
      DO 401 I=1,2
      I1=INDTJ(J)+I-1
      I2=2*(J-1)+I
      A6=CEXP(U*BK*(SN*(XW1(I1)*CSPHI+YW1(I1)*SNPHI)+ZW1(I1)*CS))
      RWT(J1)=RWT(J1)+(CS*(DXW(I1)*CSPHI+DYW(I1)*SNPHI)-
     1 DZW(I1)*SN)*TJ(I2)*A6
      RWP(J1)=RWP(J1)+(-DXW(I1)*SNPHI+DYW(I1)*CSPHI)*TJ(I2)*A6
  401 CONTINUE
      DO 402 JJ=1,4
      ALPHA=PI*((JJ-1)/2.+0.25)
      SA=SIN(ALPHA)
      CA=COS(ALPHA)
      RA=(RADJ(J)+RADD(J))/2.
      XA=XJ(J)+RA*(CA*UXJ1(J)+SA*UXJ2(J))
      YA=YJ(J)+RA*(CA*UYJ1(J)+SA*UYJ2(J))
      ZA=ZJ(J)+RA*(CA*UZJ1(J)+SA*UZJ2(J))
      DT=CA*(UXJ1(J)*CS*CSPHI+UYJ1(J)*CS*SNPHI-UZJ1(J)*SN)+
     1 SA*(UXJ2(J)*CS*CSPHI+UYJ2(J)*CS*SNPHI-UZJ2(J)*SN)
      DP=CA*(-UXJ1(J)*SNPHI+UYJ1(J)*CSPHI)+
     1 SA*(-UXJ2(J)*SNPHI+UYJ2(J)*CSPHI)
      RR=(RADJ(J)-RADD(J))/2.
      A6=CEXP(U*BK*(SN*(XA*CSPHI+YA*SNPHI)+ZA*CS))
      RWT(J1)=RWT(J1)+RR/4.*DT*A6
      RWP(J1)=RWP(J1)+RR/4.*DP*A6
```

```
10870  402  CONTINUE
10880  400  CONTINUE
10890   156 CONTINUE
10900       RETURN
10910       END
10920C ***********************************************************
10930       SUBROUTINE LPCUR(NMODE,NM,CB,LW,NJ,CW,LC,LE,CC)
10940C ***********************************************************
10950C LIST AND PLOT BOT, WIRE, AND CAP CURRENTS.
10960       COMPLEX U,A3,A4,CB(1),CW(1),CC(1)
10970       DIMENSION TMAG(41),TPHS(41),ZMAG(41),ZPHS(41)
10980       DATA PI,U /3.14159265,(0.,1.)/
10990       WRITE(6,3)
11000  3    FORMAT(16H1   BOT CURRENTS)
11010       KM=2*NMODE-1
11020       LS=NM*2
11030       DO 100 MM=1,KM
11040       M=MM-NMODE
11050       I1=(MM-1)*LS+1
11060       I2=I1+NM-1
11070       WRITE(6,1) M,(CB(I),I=I1,I2)
11080  1    FORMAT(/,29H T-DIRECTED CURRENTS FOR MODE,I3,/,/,4(2E12.4,5X))
11090       I1=I2+1
11100       I2=I1+NM-1
11110       WRITE(6,2) M,(CB(I),I=I1,I2)
11120  2    FORMAT(/,29H Z-DIRECTED CURRENTS FOR MODE,I3,/,/,4(2E12.4,5X))
11130 100   CONTINUE
11140       DO 320 I=1,NM
11150       DO 310 K=1,41
11160       A3=0.0
11170       A4=0.0
11180       ZE=-1.0+(K-1)*0.05
11190       M=-NMODE
11200       I1=I-LS
11210       DO 305 MM=1,KM
11220       M=M+1
11230       I1=I1+LS
11240       I2=I1+NM
11250C SUM THE T AND Z DIRECTED CURRENTS ON TRIANGLE FUNCTION I AT Z=ZE,
11260C (EQ. 8).
11270       A3=A3+CB(I1)*CEXP(U*M*PI*ZE)
11280       A4=A4+CB(I2)*(CEXP(U*M*PI*ZE)-(-1)**M)
11290 305   CONTINUE
11300       TMAG(K)=CABS(A3)/2.
11310       TPHS(K)=ATAN2(AIMAG(A3),REAL(A3))*180./PI
11320       ZMAG(K)=CABS(A4)/2.
11330       ZPHS(K)=ATAN2(AIMAG(A4),REAL(A4))*180./PI
11340 310   CONTINUE
11350       WRITE(6,301)
11360 301   FORMAT(1H1)
11370       WRITE(6,302) I
11380 302   FORMAT(41H T-DIRECTED CURRENTS ON TRIANGLE FUNCTION,I3,/)
```

```
C PLOT MAGNITUDE AND PHASE.
      CALL PLOTC(TMAG,TPHS)
      WRITE(6,304) I
304   FORMAT(///,41H Z-DIRECTED CURRENTS ON TRIANGLE FUNCTION,I3,/)
C PLOT MAGNITUDE AND PHASE.
      CALL PLOTC(ZMAG,ZPHS)
320   CONTINUE
      IF(LW.EQ.0) GO TO 500
      WRITE(6,4)
4     FORMAT(///,16H   WIRE CURRENTS,/)
      WRITE(6,5) (CW(I),I=1,LW)
5     FORMAT(4(2E12.4,5X))
500   CONTINUE
      IF(NJ.EQ.0) GO TO 600
      I1=LW+1
      I2=LW+NJ
      WRITE(6,6)
6     FORMAT(///,20H   JUNCTION CURRENTS,/)
      WRITE(6,5) (CW(I),I=I1,I2)
600   CONTINUE
      IF(LC.EQ.0) RETURN
      WRITE(6,7)
7     FORMAT(///,15H   CAP CURRENTS,/)
      WRITE(6,8)
8     FORMAT(/,11H T-DIRECTED)
      WRITE(6,5) (CC(I),I=1,LC)
      WRITE(6,9)
9     FORMAT(/,11H R-DIRECTED)
      I1=LC+1
      I2=2*LC
      WRITE(6,5) (CC(I),I=I1,I2)
      IF(LE.EQ.0) RETURN
      WRITE(6,10)
10    FORMAT(//,14H EDGE CURRENTS)
      I1=2*LC+1
      I2=2*LC+LE
      WRITE(6,5) (CC(I),I=I1,I2)
      RETURN
      END
C-----------------------------------------------------------
      SUBROUTINE PLOTC(Y1,Y2)
C*************************************************************
C
C WRITTEN 2/14/74   BY J. M. PUTNAM       DEPT 220     X23877
C
C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C    N IS THE NUMBER OF POINTS TO BE PLOTTED.
C    NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C    NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C       NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C
      REAL Y1(41),Y2(41),HEAD(10)
```

```
11900       INTEGER  LINE(101),BLANK,STAR,PLUS
11910       DATA  BLANK,STAR,PLUS /1H ,1H*,1H+/
11920       N=41
11930       NR=21
11940       NC=41
11950       N10=(NC-1)/10
11960       WRITE(6,500)
11970  500  FORMAT(12H * MAGNITUDE,45X,7H+ PHASE)
11980       WRITE(6,504)
11990       XMIN=-1.0
12000       XMAX=1.0
12010       YMIN1=Y1(1)
12020       YMAX1=Y1(1)
12030       YMIN2=-180.0
12040       YMAX2=180.0
12050       DO 6 I=1,N
12060       IF(Y1(I).LT.YMIN1) YMIN1=Y1(I)
12070       IF(Y1(I).GT.YMAX1) YMAX1=Y1(I)
12080  6    CONTINUE
12090       DO 5 I =1,N10
12100       Z=I
12110     5 HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
12120       DY1=(YMAX1-YMIN1)/(NR-1)
12130       DY2=18.0
12140       Z1=YMAX1+DY1
12150       Z2=YMAX2+DY2
12160       YL1=Z1-DY1/2.
12170       YL2=Z2-DY2/2.
12180       DO 7 J=1,NR
12190       DO 8 K=1,NC
12200  8    LINE(K)=BLANK
12210       Z1=Z1-DY1
12220       Z2=Z2-DY2
12230       YU1=YL1
12240       YU2=YL2
12250       YL1=Z1-DY1/2.
12260       YL2=Z2-DY2/2.
12270       DO 9 I=1,N
12280       X=XMIN+(XMAX-XMIN)*(I-1)/FLOAT(N-1)
12290       IF(Y2(I).GE.YU2) GO TO 10
12300       IF(Y2(I).LT.YL2) GO TO 10
12310       K=(X-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
12320       IF(K.GT.NC)  K=NC
12330       LINE(K)=PLUS
12340  10   IF(Y1(I).GE.YU1) GO TO 9
12350       IF(Y1(I).LT.YL1)  GO TO 9
12360       K=(X-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
12370       IF(K.GT.NC)  K=NC
12380       LINE(K)=STAR
12390  9    CONTINUE
12400       WRITE(6,508) Z1,(LINE(K),K=1,NC),Z2
12410  7    CONTINUE
```

```
      WRITE(6,504)
      WRITE( 6,3002)
      WRITE( 6,507) XMIN,(HEAD(I),I=1,N10)
      RETURN
  504 FORMAT ( 1X, 14(1H-), 1H., 8(5H----.), 1H- )
  507 FORMAT(3X,3HZ/L,5X,11(G10.2))
  508 FORMAT (1X, G12.3,1X, 1HI, 41A1, 1HI,F7.1)
 3002 FORMAT(15X,1HI,47(1H-),1HI)
      END
C -------------------------------------------------
      SUBROUTINE NEARB(XTEST,YTEST,ZTEST,ZM)
C CALCULATE THE M-TH MODAL CURRENT COEFFICIENTS FOR THE NEAR FIELD
C ANALYSIS (EQS 67 & 71).
      COMPLEX U,A3,ZM(1)
      COMPLEX GT(82),GZ(82),G1T(82),G1Z(82),H1T(82)
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
      COMMON /BOT3/ DH(82),SV(82),CV(82)
      COMMON /BOT5/ T(160),TP(160),TZ(160)
      COMMON /INT/ M,RHO2,ZP
      EXTERNAL FUNCT,FUNCZ,FUNC1T,FUNC1Z,FUNC2T
      DATA PI,U /3.14159265,(0.,1.)/
      BKL=BK*BL
      KG=NP-1
      NM=(NP-3)/2
      ETA=376.707
      I1=0
C COMPUTATION OF GREEN FUNCTION KERNELS G, H0, AND H1.
      DO 16 J=1,KG
      I1=I1+1
      YY=YB1(J)-YTEST
      XX=XB1(J)-XTEST
      RHO2=XX*XX+YY*YY
      ZP=ZTEST
      CALL CSIMP(FUNCT,-BL,BL,0.05,10,A3,GT(I1),K,IER)
      GT(I1)=GT(I1)/4./PI
      CALL CSIMP(FUNCZ,-BL,BL,0.05,10,A3,GZ(I1),K,IER)
      GZ(I1)=GZ(I1)/4./PI
      CALL CSIMP(FUNC1T,-BL,BL,0.05,10,A3,G1T(I1),K,IER)
      G1T(I1)=G1T(I1)/4./PI
      CALL CSIMP(FUNC1Z,-BL,BL,0.05,10,A3,G1Z(I1),K,IER)
      G1Z(I1)=G1Z(I1)/4./PI
      CALL CSIMP(FUNC2T,-BL,BL,0.05,10,A3,H1T(I1),K,IER)
      H1T(I1)=H1T(I1)/4./PI
   16 CONTINUE
C COMPUTATION OF M-TH MODAL CURRENT COEFFICIENTS (EQS. 67 & 71).
      DO 30 J=1,NM
      JL=(J-1)*6
      J3=(J-1)*4
      J1=2*(J-1)
C ELECTRIC FIELD.
      L1=JL+1
```

```
12940       L2=L1+NM*6
12950       L3=L1+1
12960       L4=L2+1
12970       L5=L1+2
12980       L6=L2+2
12990       ZM(L1)=0.
13000       ZM(L2)=0.
13010       ZM(L3)=0.
13020       ZM(L4)=0.
13030       ZM(L5)=0.
13040       ZM(L6)=0.
13050C MAGNETIC FIELD.
13060       K1=L1+3
13070       K2=L2+3
13080       K3=L1+4
13090       K4=L2+4
13100       K5=L1+5
13110       K6=L2+5
13120       ZM(K1)=0.
13130       ZM(K2)=0.
13140       ZM(K3)=0.
13150       ZM(K4)=0.
13160       ZM(K5)=0.
13170       ZM(K6)=0.
13180       DO 70 JJ=1,4
13190       J2=J1+JJ
13200       J7=J3+JJ
13210       J4=J2
13220       XX=XB1(J2)-XTEST
13230       YY=YB1(J2)-YTEST
13240C ELECTRIC FIELD.
13250C UX COMPONENT.
13260       ZM(L1)=ZM(L1)+DH(J2)*(CV(J2)*T(J7)*GT(J4)+
13270      1 XX*TP(J7)*G1T(J4)/BK/BK)
13280       ZM(L2)=ZM(L2)+DH(J2)*XX*TZ(J7)*G1T(J4)
13290C UY COMPONENT.
13300       ZM(L3)=ZM(L3)+DH(J2)*(SV(J2)*T(J7)*GT(J4)+
13310      1 YY*TP(J7)*G1T(J4)/BK/BK)
13320       ZM(L4)=ZM(L4)+DH(J2)*YY*TZ(J7)*G1T(J4)
13330C UZ COMPONENT.
13340       ZM(L5)=ZM(L5)+DH(J2)*TP(J7)*H1T(J4)
13350       ZM(L6)=ZM(L6)+DH(J2)*TZ(J7)*(GZ(J4)+
13360      1 U*R*PI*H1T(J4)/BK/BK)
13370C MAGNETIC FIELD.
13380C UX COMPONENT.
13390       ZM(K1)=ZM(K1)-DH(J2)*SV(J2)*T(J7)*H1T(J4)
13400       ZM(K2)=ZM(K2)+DH(J2)*YY*TZ(J7)*G1Z(J4)
13410C UY COMPONENT.
13420       ZM(K3)=ZM(K3)+DH(J2)*CV(J2)*T(J7)*H1T(J4)
13430       ZM(K4)=ZM(K4)-DH(J2)*XX*TZ(J7)*G1Z(J4)
13440C UZ COMPONENT.
13450       ZM(K5)=ZM(K5)-DH(J2)*T(J7)*(YY*CV(J2)-
```

```
13460      1 XX*SV(J2))*G1T(J4)
13470    70 CONTINUE
13480C MULTIPLY BY CONSTANTS.
13490      ZM(L1)=U*BK*ETA*ZM(L1)
13500      ZM(L2)=-M*PI*ETA/BKL*ZM(L2)
13510      ZM(L3)=U*BK*ETA*ZM(L3)
13520      ZM(L4)=-M*PI*ETA/BKL*ZM(L4)
13530      ZM(L5)=U*ETA/BK*ZM(L5)
13540      ZM(L6)=U*BK*ETA*ZM(L6)
13550    30 CONTINUE
13560      RETURN
13570      END
13580C ------------------------------------------------
13590      SUBROUTINE NEARC(XTEST,YTEST,ZTEST,ZM)
13600C CALCULATE THE CAP CURRENT COEFFICIENTS FOR THE NEAR FIELD ANALYSIS.
13610      COMPLEX U,A3,A4,ZM(1)
13620      COMMON /WAVE/ BK
13630      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
13640      COMMON /BOT3/ DH(82),SV(82),CV(82)
13650      COMMON /BOT5/ T(160),TP(160),TZ(160)
13660      COMMON /CAP1/ NC,XC,YC,ZC(2)
13670      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
13680      COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
13690      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
13700      DATA PI,U /3.14159265,(0.,1.)/
13710      NM=(NP-3)/2
13720      LR=(NPR-3)/2
13730      LC=NC*NM*LR
13740      BK2=BK*BK
13750      ETA=376.707
13760      J=0
13770C NOTE, EDGE COEFFICIENTS ARE NOT INCLUDED.
13780      DO 30 JC=1,NC
13790      DO 30 JP=1,NM
13800      DO 30 JR=1,LR
13810      J=J+1
13820      JL=(J-1)*6
13830C ELECTRIC FIELD.
13840      L1=JL+1
13850      L2=L1+LC*6
13860      L3=L1+1
13870      L4=L2+1
13880      L5=L1+2
13890      L6=L2+2
13900      ZM(L1)=0.
13910      ZM(L2)=0.
13920      ZM(L3)=0.
13930      ZM(L4)=0.
13940      ZM(L5)=0.
13950      ZM(L6)=0.
13960C MAGNETIC FIELD.
13970      K1=L1+3
```

```
      K2=L2+3
      K3=L1+4
      K4=L2+4
      K5=L1+5
      K6=L2+5
      ZM(K1)=0.
      ZM(K2)=0.
      ZM(K3)=0.
      ZM(K4)=0.
      ZM(K5)=0.
      ZM(K6)=0.
      JP2=2*(JP-1)
      JP7=4*(JP-1)
      DO 70 JJP=1,4
      JP2=JP2+1
      JP7=JP7+1
      JR2=2*(JR-1)
      JR7=4*(JR-1)
      DO 70 JJR=1,4
      JR2=JR2+1
      JR7=JR7+1
      AJ=AC(JP2)*ABS(RHOC(JR2+1)**2-RHOC(JR2)**2)
      RJ=RHOC1(JR2)*RC1(JP2)
      XX=XC+RJ*CPC(JP2)-XTEST
      YY=YC+RJ*SPC(JP2)-YTEST
      ZZ=ZC(JC)-ZTEST
      RR=SQRT(XX*XX+YY*YY+ZZ*ZZ)
      A3=CEXP(-U*BK*RR)/4./PI/RR
      A4=(1./RR+U*BK)/RR*A3
C  ELECTRIC FIELD.
C  UX COMPONENT.
       ZM(L1)=ZM(L1)+AJ*TCT(JR7)*(CV(JP2)*T(JP7)*A3+
     1  XX/BK2*TP(JP7)/RHOC1(JR2)*A4)
       ZM(L2)=ZM(L2)+AJ*TZ(JP7)*(CPC(JP2)*TCR(JR7)*A3+
     1  XX/BK2*(TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)*A4)
C  UY COMPONENT.
       ZM(L3)=ZM(L3)+AJ*TCT(JR7)*(SV(JP2)*T(JP7)*A3+
     1  YY/BK2*TP(JP7)/RHOC1(JR2)*A4)
       ZM(L4)=ZM(L4)+AJ*TZ(JP7)*(SPC(JP2)*TCR(JR7)*A3+
     1  YY/BK2*(TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)*A4)
C  UZ COMPONENT.
       ZM(L5)=ZM(L5)+AJ*ZZ*TP(JP7)/RHOC1(JR2)*TCT(JR7)*A4
       ZM(L6)=ZM(L6)+AJ*ZZ*TZ(JP7)*
     1  (TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)*A4
C  MAGNETIC FIELD.
C  UX COMPONENT.
       ZM(K1)=ZM(K1)-AJ*ZZ*SV(JP2)*T(JP7)*TCT(JR7)*A4
       ZM(K2)=ZM(K2)-AJ*ZZ*SPC(JP2)*TZ(JP7)*TCR(JR7)*A4
C  UY COMPONENT.
       ZM(K3)=ZM(K3)+AJ*ZZ*CV(JP2)*T(JP7)*TCT(JR7)*A4
       ZM(K4)=ZM(K4)+AJ*ZZ*CPC(JP2)*TZ(JP7)*TCR(JR7)*A4
C  UZ COMPONENT.
```

```
      ZM(K5)=ZM(K5)-AJ*T(JP7)*TCT(JR7)*
     1 (YY*CV(JP2)-XX*SV(JP2))*A4
      ZM(K6)=ZM(K6)-AJ*TZ(JP7)*TCR(JR7)*
     1 (YY*CPC(JP2)-XX*SPC(JP2))*A4
   70 CONTINUE
C MULTIPLY BY CONSTANTS.
      ZM(L1)=U*BK*ETA*ZM(L1)
      ZM(L2)=U*BK*ETA*ZM(L2)
      ZM(L3)=U*BK*ETA*ZM(L3)
      ZM(L4)=U*BK*ETA*ZM(L4)
      ZM(L5)=U*ETA/BK*ZM(L5)
      ZM(L6)=U*ETA/BK*ZM(L6)
   30 CONTINUE
      RETURN
      END
C -------------------------------------------------
      SUBROUTINE NEARW(XTEST,YTEST,ZTEST,ZM)
C CALCULATE THE WIRE/JUNCTION CURRENT COEFFICIENTS FOR THE NEAR
C FIELD ANALYSIS.
      COMPLEX U,A3,A4,ZM(1)
      COMMON /WAVE/ BK
      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
     1 XW1(100),YW1(100),ZW1(100)
      COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
      COMMON /WIRE3/ NW,INDW(6),RADW(5)
      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(+9)
      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
      COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
      DATA PI,U /3.14159265,(0.,1.)/
      LWJ=LW+NJ
      BK2=BK*BK
      ETA=376.707
C NOTE, JUNCTION COEFFICIENTS ARE NOT INCLUDED.
      DO 30 J=1,LW
      JL=(J-1)*6
      J3=(J-1)*4
      J1=INDTW(J)-1
C ELECTRIC FIELD.
      L1=JL+1
      L3=L1+1
      L5=L1+2
      ZM(L1)=0.
      ZM(L3)=0.
      ZM(L5)=0.
C MAGNETIC FIELD.
      K1=L1+3
      K3=L1+4
      K5=L1+5
```

```
15010       ZM(K1)=0.
15020       ZM(K3)=0.
15030       ZM(K5)=0.
15060       DO 70 JJ=1,4
15070       J2=J1+JJ
15080       J7=J3+JJ
15090       XX=XW1(J2)-XTEST
15100       YY=YW1(J2)-YTEST
15110       ZZ=ZW1(J2)-ZTEST
15120       RR=SQRT(XX*XX+YY*YY+ZZ*ZZ)
15130       A3=CEXP(-U*BK*RR)/4./PI/RR
15140       A4=(1./RR+U*BK)/RR*A3
15150C ELECTRIC FIELD.
15160C UX COMPONENT.
15170       ZM(L1)=ZM(L1)+DXW(J2)*TW(J7)*A3+
15180      1 DHW(J2)*XX*TPW(J7)/BK2*A4
15190C UY COMPONENT.
15200       ZM(L3)=ZM(L3)+DYW(J2)*TW(J7)*A3+
15210      1 DHW(J2)*YY*TPW(J7)/BK2*A4
15220C UZ COMPONENT.
15230       ZM(L5)=ZM(L5)+DZW(J2)*TW(J7)*A3+
15240      1 DHW(J2)*ZZ*TPW(J7)/BK2*A4
15250C MAGNETIC FIELD.
15260C UX COMPONENT.
15270       ZM(K1)=ZM(K1)-(DYW(J2)*ZZ-
15280      1 DZW(J2)*YY)*TW(J7)*A4
15290C UY COMPONENT.
15300       ZM(K3)=ZM(K3)-(DZW(J2)*XX-
15310      1 DXW(J2)*ZZ)*TW(J7)*A4
15320C UZ COMPONENT.
15330       ZM(K5)=ZM(K5)-(DXW(J2)*YY-
15340      1 DYW(J2)*XX)*TW(J7)*A4
15350    70 CONTINUE
15360C MULTIPLY BY CONSTANTS.
15370       ZM(L1)=U*BK*ETA*ZM(L1)
15380       ZM(L3)=U*BK*ETA*ZM(L3)
15390       ZM(L5)=U*BK*ETA*ZM(L5)
15400    30 CONTINUE
15410       RETURN
15420       END
15430C ***********************************************************
15440       SUBROUTINE CSIMP(F,A,B,DEL,IMAX,SI1,S,N,IER)
15450C ***********************************************************
15460C COMPLEX SIMPSON INTEGRATION ROUTINE.
15470       COMPLEX F,SI1,S,SUMK
15480       SI1=(.0,.0)
15490       S=(.0,.0)
15500       N=0
15510       BA=B-A
15520       IF (BA)20,19,20
15530    19 IER=1
```

```
15540       RETURN
15550    20 IF(DEL)22,22,23
15560    22 IER=2
15570       RETURN
15580    23 IF(IMAX-1)24,24,25
15590    24 IER=3
15600       RETURN
15610    25 X=BA/2.+A
15620       NHALF=1
15630       SUMK=F(X)*BA*2./3.
15640       S=SUMK+(F(A)+F(B))*BA/6.
15650       DO 28 I=2,IMAX
15660       SI1=S
15670       S=(S-SUMK/2.)/2.
15680       NHALF=NHALF*2
15690       ANHLF=NHALF
15700       FRSTX=A+(BA/ANHLF)/2.
15710       SUMK=F(FRSTX)
15720       XK=FRSTX
15730       KLAST=NHALF-1
15740       FINC=BA/ANHLF
15750       DO 26 K=1,KLAST
15760       XK=XK+FINC
15770    26 SUMK=SUMK+F(XK)
15780       SUMK=SUMK*2.*BA/(3.*ANHLF)
15790       S=S+SUMK
15800       IF(CABS(S).EQ.0.0) GO TO 29
15810       IF((CABS(S-SI1)/CABS(S))-DEL) 29,28,28
15820    28 CONTINUE
15830       IER=4
15840       WRITE(6,1) SI1,S
15850  1    FORMAT(///,29H INTEGRATION DID NOT CONVERGE,//
15860      1 50H THE PREVIOUS AND FINAL VALUES OF THE INTEGRAL ARE,
15870      2 13H RESPECTIVELY,4E12.3)
15880       GO TO 30
15890    29 IER=0
15900    30 N=2*NHALF
15910       RETURN
15920       END
15930C ------------------------------------------------
15940       COMPLEX FUNCTION FUNCT(Z)
15950       COMPLEX U
15960       COMMON /WAVE/ BK
15970       COMMON /BOT2/ NP,BL
15980       COMMON /INT/ M,RHO2,ZP
15990       DATA PI,U /3.14159265,(0.,1.)/
16000       R=SQRT(RHO2+(Z-ZP)**2)
16010       FUNCT=CEXP(U*(M*PI*Z/BL-BK*R))/R
16020       RETURN
16030       END
16040C ------------------------------------------------
16050       COMPLEX FUNCTION FUNCZ(Z)
```

```
      COMPLEX U
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL
      COMMON /INT/ M,RHO2,ZP
      DATA PI,U /3.14159265,(0.,1.)/
      R=SQRT(RHO2+(Z-ZP)**2)
      FUNCZ=(CEXP(U*M*PI*Z/BL)-(-1)**M)*CEXP(-U*BK*R)/R
      RETURN
      END
C -------------------------------------------------
      COMPLEX FUNCTION FUNC1T(Z)
      COMPLEX U
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL
      COMMON /INT/ M,RHO2,ZP
      DATA PI,U /3.14159265,(0.,1.)/
      R=SQRT(RHO2+(Z-ZP)**2)
      FUNC1T=CEXP(U*(M*PI*Z/BL-BK*R))*(1./R+U*BK)/R/R
      RETURN
      END
C -------------------------------------------------
      COMPLEX FUNCTION FUNC1Z(Z)
      COMPLEX U
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL
      COMMON /INT/ M,RHO2,ZP
      DATA PI,U /3.14159265,(0.,1.)/
      R=SQRT(RHO2+(Z-ZP)**2)
      FUNC1Z=(CEXP(U*M*PI*Z/BL)-(-1)**M)*CEXP(-U*BK*R)*
     1 (1./R+U*BK)/R/R
      RETURN
      END
C -------------------------------------------------
      COMPLEX FUNCTION FUNC2T(Z)
      COMPLEX U
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL
      COMMON /INT/ M,RHO2,ZP
      DATA PI,U /3.14159265,(0.,1.)/
      R=SQRT(RHO2+(Z-ZP)**2)
      FUNC2T=CEXP(U*(M*PI*Z/BL-BK*R))*(Z-ZP)*(1./R+U*BK)/R/R
      RETURN
      END
C -------------------------------------------------
      SUBROUTINE PLOT(X,Y,ISYM,N,LX,LY,NC,NR)
C -------------------------------------------------
C
C WRITTEN 2/14/74   BY J. M. PUTNAM        DEPT 220      X23877
C
C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C    N IS THE NUMBER OF POINTS TO BE PLOTTED.
C    NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
```

```
16580C     NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
16590C        NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
16600C
16610       DIMENSION X(1),Y(1),ISYM(1),HEAD(10)
16620       INTEGER  LINE(102),BLANK,PLUS
16630       DATA BLANK,PLUS /1H ,1H+/
16640       NC1=NC+1
16650       WRITE(6,501)
16660  501  FORMAT(1H1)
16670       N10=(NC-1)/10
16680       XMIN=X(1)
16690       XMAX=X(1)
16700       YMIN=Y(1)
16710       YMAX=Y(1)
16720       DO 6 I=1,N
16730       IF(X(I).LT.XMIN) XMIN=X(I)
16740       IF(X(I).GT.XMAX) XMAX=X(I)
16750       IF(Y(I).LT.YMIN) YMIN=Y(I)
16760       IF(Y(I).GT.YMAX) YMAX=Y(I)
16770  6    CONTINUE
16780       DELX=(XMAX-XMIN)/(NC-1)
16790       DELY=(YMAX-YMIN)/(NR-1)
16800       DO 5 I=1,N10
16810  5    HEAD(I)=XMIN+10*I*DELX
16820       DO 11 K=1,NC1
16830       LINE(K)='-'
16840       IF(MOD(K,5).EQ.1) LINE(K)='.'
16850  11   CONTINUE
16860       WRITE(6,504) (LINE(K),K=1,NC1)
16870       YU=YMAX+DELY/2.
16880       YL=YMAX-DELY/2.
16890       DO 7 J=1,NR
16900       DO 8 K=1,NC
16910  8    LINE(K)=BLANK
16920       LINE(NC1)='I'
16930       DO 9 I=1,N
16940       IF(Y(I).GE.YU) GO TO 9
16950       IF(Y(I).LT.YL) GO TO 9
16960       K=(X(I)-XMIN)/DELX+1.5
16970       IF(LINE(K).NE.BLANK) LINE(K)=PLUS
16980       IF(LINE(K).EQ.BLANK) LINE(K)=ISYM(I)
16990  9    CONTINUE
17000       YM=(YU+YL)/2.
17010       WRITE(6,508) YM,(LINE(K),K=1,NC1)
17020       YU=YL
17030       YL=YL-DELY
17040  7    CONTINUE
17050       DO 12 K=1,NC1
17060       LINE(K)='-'
17070       IF(MOD(K,5).EQ.1) LINE(K)='.'
17080  12   CONTINUE
17090       WRITE(6,504) (LINE(K),K=1,NC1)
```

```
      DO 13 K=1,NC1
      LINE(K)=BLANK
      IF(MOD(K,10).EQ.1) LINE(K)='I'
13    CONTINUE
      WRITE( 6,3002) LY,LX,(LINE(K),K=1,NC1)
      WRITE( 6,507) XMIN,(HEAD(I),I=1,N10)
      WRITE(6,502) N
502   FORMAT(/,2X,I4,15H POINTS PLOTTED)
      RETURN
504   FORMAT(1X,14(1H-),102A1)
 507  FORMAT(10X,11(F10.4))
 508  FORMAT (1X, F12.4,1X, 1HI, 102A1 )
3002  FORMAT(4X,A1,54  /  ,A1,4X,102A1)
      END
/EOR
```

BOTSCM

```
00100BOTS,P30,T100,CM140000,STCX3.
00110ACCOUNT,NO326S,BOT2A.
00120BANNERS(OUTPUT) *J. PUTNAM*DEPT 220*BLD 110-4**
00130ATTACH(INFIL=YSSX)
00140FTN(R=0,OPT=1)
00164LDSET(PRESET=INDEF)
00170LGO.
00190/EOR
00200       PROGRAM BOTSCM(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
00210      1 INFIL,TAPE1=INFIL)
00220C
00230C  BOTSCM IS THE BOT MONO-STATIC SCATTERING CODE.
00240C
00250C  UNIT 5 IS THE CARD READER.
00260C  UNIT 6 IS THE LINE PRINTER.
00270C  UNIT 1 IS A DISK FILE CONTAINING THE Y MATRIX.
00280C
00285       COMPLEX U
00300       COMPLEX STT(91),SPP(91),STP(91),SPT(91)
00305       COMPLEX CBT(82),CBP(82),CWT(59),CWP(59),CCT(96),CCP(96)
00320       COMPLEX Y(2304)
00325       COMPLEX RBT(736),RBP(736),RWT(368),RWP(368)
00326       COMPLEX RCT(2208),RCP(2208)
00327       COMMON /WAVE/ BK
00330       COMMON /BOT1/ NMODE,NPT,NBAND
00340       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
00350       COMMON /BOT3/ DH(82),SV(82),CV(82)
00370       COMMON /BOT5/ T(160),TP(160),TZ(160)
00380       COMMON /BOT6/ IEDGE,IUNIF
00382       COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
00384      1 XW1(100),YW1(100),ZW1(100)
00385       COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
00386       COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
00387       COMMON /CAP1/ NC,XC,YC,ZC(2)
00388       COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
00389       COMMON /EDG1/ NE,ZE(2),ZBE(10)
00400       DIMENSION THS(73),PHIS(73)
00410       DIMENSION ANG(6),IPLANE(6),ANG1(6),ANG2(6)
00420       DIMENSION MORD(2)
00430       U=(0.,1.)
00440       ETA=376.707
00450       PI=3.14159265
00460       DTOR=PI/180.
00462       READ(5,1) BK
00463  1    FORMAT(E15.7)
00464       WRITE(6,2) BK
00465  2    FORMAT(9H1       BK,/,E15.7)
00470C
00480       CALL BOTIN
00490C
00500       CALL CAPIN
00510C
```

```
00520       CALL WIREIN
00522C
00530C READ INPUTS, DESCIBING THE SCATTERING PLANES.
00540       READ(5,44) NANG,NT,PHII,THI
00550 44    FORMAT(2I3,2F8.4)
00590       READ(5,53) (ANG(I),I=1,NANG)
00600 53    FORMAT(10F8.4)
00610C IPLANE = 1 FOR PHI FIXED.
00620C          2 FOR THETA FIXED.
00630       READ(5,54) (IPLANE(I),I=1,NANG)
00640 54    FORMAT(10I8)
00642       READ(5,53) (ANG1(I),I=1,NANG)
00644       READ(5,53) (ANG2(I),I=1,NANG)
00650       WRITE(6,3) NANG,NT
00660 3     FORMAT(///,25H NUMBER OF FIXED ANGLES =,I3,//
00670      1 35H NUMBER OF ANGLES PER FIXED ANGLE =,I4)
00680       WRITE(6,4) (ANG(I),IPLANE(I),ANG1(I),ANG2(I),I=1,NANG)
00690 4     FORMAT(///,45H FIXED ANGLE     CODE      VARIABLE ANGLE RANGE,
00700      1 //,(F8.1,9X,I2,6X,F8.1,3H - ,F8.1))
00710       LS=NP-3
00712       NM=LS/2
00714       NM1=NM+1
00716       NM4=NM*4
00720       LSS=LS*LS
00722       LR=(NPR-3)/2
00724       LC=NC*NM*LR
00726       LC2=LC*2
00728       LE=NE*NM
00730       LSC=LS*(LC2+LE)
00740       LSW=LS*(LW+NJ)
00750       LCC=(LC2+LE)**2
00760       LCW=(LC2+LE)*(LW+NJ)
00770       LWW=(LW+NJ)**2
00810       BKL=BK*BL
00811       P1=(BK*BK*ETA/2./PI)**2/4./PI
00812       KMODE=2*NMODE-1
00813       CALL ORDER(MORD)
00814       DO 90 IANG=1,NANG
00815       REWIND 1
00816       READ(1) I
00817       READ(1) I
00818C INITIALIZE ARRAYS, AND SET THE SCATTERING ANGLES.
00819       IPLAN=IPLANE(IANG)
00840       DT=(ANG2(IANG)-ANG1(IANG))/(NT-1)
00885       DO 100 K=1,NT
00886       PHIS(K)=ANG(IANG)*(2-IPLAN)+(ANG1(IANG)+DT*(K-1))*(IPLAN-1)
00887       THS(K)=ANG(IANG)*(IPLAN-1)+(ANG1(IANG)+DT*(K-1))*(2-IPLAN)
00888       STT(K)=0.0
00889       SPP(K)=0.0
00890       STP(K)=0.0
00891 100   SPT(K)=0.0
00892       CALL RBOT(RBT,RBP,NT,THS,PHIS)
```

```
00894       IF(NC.NE.0) CALL RCAP(RCT,RCP,NT,THS,PHIS)
00896       IF(NPW.NE.0) CALL RWIRE(RWT,RWP,NT,THS,PHIS)
00900       NUMB=0
00910 50    CONTINUE
00920       READ(1) M,N
00930       IF(EOF(1)) 37,60
00931 60    NUMB=NUMB+1
00932       DO 120 K=1,NT
00933       KB=(K-1)*LS
00934       KC=(K-1)*(LC2+LE)
00935       KW=(K-1)*(LW+NJ)
00940       DMT=2.*BL*SINC(M*PI+BKL*COS(THS(K)*DTOR))
00942       DMZ=DMT-2.*BL*(-1)**M*SINC(BKL*COS(THS(K)*DTOR))
00950C DETERMINE SUBMATRIX TYPE, AND CALCULATE CURRENTS.
00960       IF(N.LT.NMODE) GO TO 61
00970       IF(N.EQ.MORD(1)) GO TO 62
00980       IF(N.EQ.MORD(2)) GO TO 63
00990       GO TO 120
00992 61    DNT=2.*BL*SINC(-N*PI+BKL*COS(THS(K)*DTOR))
00994       DNZ=DNT-2.*BL*(-1)**N*SINC(BKL*COS(THS(K)*DTOR))
01010       IF(M.LT.NMODE) GO TO 71
01020       IF(M.EQ.MORD(1)) GO TO 72
01030       IF(M.EQ.MORD(2)) GO TO 73
01040       GO TO 120
01050 62    IF(M.LT.NMODE) GO TO 74
01060       IF(M.EQ.MORD(1)) GO TO 75
01070       IF(M.EQ.MORD(2)) GO TO 76
01080       GO TO 120
01090 63    IF(M.LT.NMODE) GO TO 77
01100       IF(M.EQ.MORD(1)) GO TO 78
01110       IF(M.EQ.MORD(2)) GO TO 79
01120       GO TO 120
01130C BOT-BOT.
01140 71    IF(K.EQ.1) READ(1) (Y(I),I=1,LSS)
01162       CALL ZERO(LS,CBT)
01164       CALL ZERO(LS,CBP)
01170       CALL MULTYR(LS,LS,Y,DNT,DNZ,RBT(KB+1),CBT)
01175       CALL MULTYR(LS,LS,Y,DNT,DNZ,RBP(KB+1),CBP)
01176       CALL MULTRC(LS,DMT,DMZ,RBT(KB+1),CBT,STT(K))
01177       CALL MULTRC(LS,DMT,DMZ,RBP(KB+1),CBP,SPP(K))
01178       CALL MULTRC(LS,DMT,DMZ,RBT(KB+1),CBP,STP(K))
01179       CALL MULTRC(LS,DMT,DMZ,RBP(KB+1),CBT,SPT(K))
01180       GO TO 120
01190C CAP-BOT.
01200 72    IF(K.EQ.1) READ(1) (Y(I),I=1,LSC)
01202       CALL ZERO(LC2+LE,CCT)
01204       CALL ZERO(LC2+LE,CCP)
01210       CALL MULTYR(LC2+LE,LS,Y,DNT,DNZ,RBT(KB+1),CCT)
01215       CALL MULTYR(LC2+LE,LS,Y,DNT,DNZ,RBP(KB+1),CCP)
01216       CALL MULTRC(LC2+LE,1.0,1.0,RCT(KC+1),CCT,STT(K))
01217       CALL MULTRC(LC2+LE,1.0,1.0,RCP(KC+1),CCP,SPP(K))
01218       CALL MULTRC(LC2+LE,1.0,1.0,RCT(KC+1),CCP,STP(K))
```

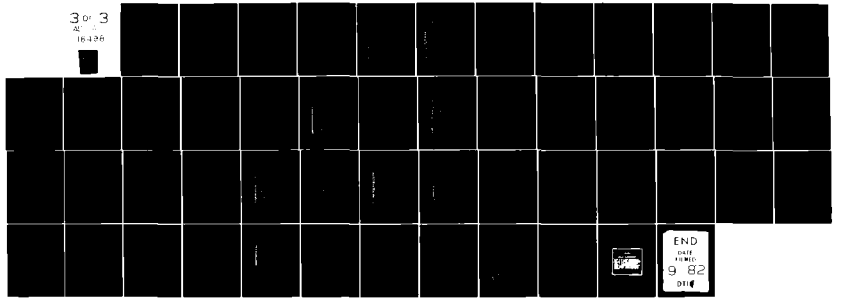
```
C1219       CALL MULTRC(LC2+LE,1.0,1.0,RCP(KC+1),CCT,SPT(K))
C1220       GO TO 120
C1230C WIRE-BOT.
C1240  73   IF(K.EQ.1) READ(1) (Y(I),I=1,LSW)
C1242       CALL ZERO(LW+NJ,CWT)
C1244       CALL ZERO(LW+NJ,CWP)
C1250       CALL MULTYR(LW+NJ,LS,Y,DNT,DNZ,RBT(KB+1),CWT)
C1255       CALL MULTYR(LW+NJ,LS,Y,DNT,DNZ,RBP(KB+1),CWP)
C1256       CALL MULTRC(LW+NJ,1.0,1.0,RWT(KW+1),CWT,STT(K))
C1257       CALL MULTRC(LW+NJ,1.0,1.0,RWP(KW+1),CWP,SPP(K))
C1258       CALL MULTRC(LW+NJ,1.0,1.0,RWT(KW+1),CWP,STP(K))
C1259       CALL MULTRC(LW+NJ,1.0,1.0,RWP(KW+1),CWT,SPT(K))
C1260       GO TO 120
C1270C BOT-CAP.
C1280  74   IF(K.EQ.1) READ(1) (Y(I),I=1,LSC)
C1282       CALL ZERO(LS,CBT)
C1284       CALL ZERO(LS,CBP)
C1290       CALL MULTYR(LS,LC2+LE,Y,1.0,1.0,RCT(KC+1),CBT)
C1295       CALL MULTYR(LS,LC2+LE,Y,1.0,1.0,RCP(KC+1),CBP)
C1296       CALL MULTRC(LS,DMT,DMZ,RBT(KB+1),CBT,STT(K))
C1297       CALL MULTRC(LS,DMT,DMZ,RBP(KB+1),CBP,SPP(K))
C1298       CALL MULTRC(LS,DMT,DMZ,RBT(KB+1),CBP,STP(K))
C1299       CALL MULTRC(LS,DMT,DMZ,RBP(KB+1),CBT,SPT(K))
C1300       GO TO 120
C1310C CAP-CAP.
C1320  75   IF(K.EQ.1) READ(1) (Y(I),I=1,LCC)
C1322       CALL ZERO(LC2+LE,CCT)
C1324       CALL ZERO(LC2+LE,CCP)
C1330       CALL MULTYR(LC2+LE,LC2+LE,Y,1.0,1.0,RCT(KC+1),CCT)
C1335       CALL MULTYR(LC2+LE,LC2+LE,Y,1.0,1.0,RCP(KC+1),CCP)
C1336       CALL MULTRC(LC2+LE,1.0,1.0,RCT(KC+1),CCT,STT(K))
C1337       CALL MULTRC(LC2+LE,1.0,1.0,RCP(KC+1),CCP,SPP(K))
C1338       CALL MULTRC(LC2+LE,1.0,1.0,RCT(KC+1),CCP,STP(K))
C1339       CALL MULTRC(LC2+LE,1.0,1.0,RCP(KC+1),CCT,SPT(K))
C1340       GO TO 120
C1350C WIRE-CAP.
C1360  76   IF(K.EQ.1) READ(1) (Y(I),I=1,LCW)
C1362       CALL ZERO(LW+NJ,CWT)
C1364       CALL ZERO(LW+NJ,CWP)
C1370       CALL MULTYR(LW+NJ,LC2+LE,Y,1.0,1.0,RCT(KC+1),CWT)
C1375       CALL MULTYR(LW+NJ,LC2+LE,Y,1.0,1.0,RCP(KC+1),CWP)
C1376       CALL MULTRC(LW+NJ,1.0,1.0,RWT(KW+1),CWT,STT(K))
C1377       CALL MULTRC(LW+NJ,1.0,1.0,RWP(KW+1),CWP,SPP(K))
C1378       CALL MULTRC(LW+NJ,1.0,1.0,RWT(KW+1),CWP,STP(K))
C1379       CALL MULTRC(LW+NJ,1.0,1.0,RWP(KW+1),CWT,SPT(K))
C1380       GO TO 120
C1390C BOT-WIRE.
C1400  77   IF(K.EQ.1) READ(1) (Y(I),I=1,LSW)
C1402       CALL ZERO(LS,CBT)
C1404       CALL ZERO(LS,CBP)
C1410       CALL MULTYR(LS,LW+NJ,Y,1.0,1.0,RWT(KW+1),CBT)
C1415       CALL MULTYR(LS,LW+NJ,Y,1.0,1.0,RWP(KW+1),CBP)
```

```
C1416       CALL MULTRC(LS,DMT,DMZ,RBT(KB+1),CBT,STT(K))
C1417       CALL MULTRC(LS,DMT,DMZ,RBP(KB+1),CBP,SPP(K))
C1418       CALL MULTRC(LS,DMT,DMZ,RBT(KB+1),CBP,STP(K))
C1419       CALL MULTRC(LS,DMT,DMZ,RBP(KB+1),CBT,SPT(K))
C1420       GO TO 120
C1430C CAP-WIRE.
C1440 78    IF(K.EQ.1) READ(1) (Y(I),I=1,LCW)
C1442       CALL ZERO(LC2+LE,CCT)
C1444       CALL ZERO(LC2+LE,CCP)
C1450       CALL MULTYR(LC2+LE,LW+NJ,Y,1.0,1.0,RWT(KW+1),CCT)
C1455       CALL MULTYR(LC2+LE,LW+NJ,Y,1.0,1.0,RWP(KW+1),CCP)
C1456       CALL MULTRC(LC2+LE,1.0,1.0,RCT(KC+1),CCT,STT(K))
C1457       CALL MULTRC(LC2+LE,1.0,1.0,RCP(KC+1),CCP,SPP(K))
C1458       CALL MULTRC(LC2+LE,1.0,1.0,RCT(KC+1),CCP,STP(K))
C1459       CALL MULTRC(LC2+LE,1.0,1.0,RCP(KC+1),CCT,SPT(K))
C1460       GO TO 120
C1470C WIRE-WIRE.
C1480 79    IF(K.EQ.1) READ(1) (Y(I),I=1,LWW)
C1482       CALL ZERO(LW+NJ,CWT)
C1484       CALL ZERO(LW+NJ,CWP)
C1490       CALL MULTYR(LW+NJ,LW+NJ,Y,1.0,1.0,RWT(KW+1),CWT)
C1495       CALL MULTYR(LW+NJ,LW+NJ,Y,1.0,1.0,RWP(KW+1),CWP)
C1496       CALL MULTRC(LW+NJ,1.0,1.0,RWT(KW+1),CWT,STT(K))
C1497       CALL MULTRC(LW+NJ,1.0,1.0,RWP(KW+1),CWP,SPP(K))
C1498       CALL MULTRC(LW+NJ,1.0,1.0,RWT(KW+1),CWP,STP(K))
C1499       CALL MULTRC(LW+NJ,1.0,1.0,RWP(KW+1),CWT,SPT(K))
C1500 120   CONTINUE
C1510       GO TO 50
C1550C ALL  Y SUBMATRICES HAVE BEEN READ.
C1560 37    CONTINUE
C1562       WRITE(6,114) NUMB
C1564 114   FORMAT(///,1X,I4,17H SUBMATRICES READ)
C1630C  *** END CALCULATION OF RCS ***
C1725C PRINT OUT RCS.
C1730       WRITE(6,403)
C1735 403   FORMAT(1H1,32X,19HMONO-STATIC RCS(DB),//,20X,45(1H-),//,
C1740      1 55H      PHI     THETA          OO              CO              OO
C1745      2 8H       OO,//,
C1750      3 55H+                            --              //              -/
C1755      4 8H       /-,/)
C1780       DO 110 K=1,NT
C1939       X2=CABS(STT(K))
C1940       X2=DBCON(P1*X2*X2)
C1945       X3=CABS(SPP(K))
C1950       X3=DBCON(P1*X3*X3)
C1955       X4=CABS(STP(K))
C1960       X4=DBCON(P1*X4*X4)
C1965       X5=CABS(SPT(K))
C1970       X5=DBCON(P1*X5*X5)
C1975       WRITE(6,113) PHIS(K),THS(K),X2,X3,X4,X5
C1980 113   FORMAT(1X,2F8.1,4F12.2)
C1990 110   CONTINUE
```

```
02000 90    CONTINUE
02640       STOP
02645       END
02821C ------------------------------------------------
02822       SUBROUTINE ZERO(N,A)
02823C A=0.
02824       COMPLEX A(1)
02825       DO 100 I=1,N
02826       A(I)=C.0
02827 100   CONTINUE
02828       RETURN
02829       END
02830C ------------------------------------------------------
02840       SUBROUTINE MULTYR(NI,NJ,Y,DT,DZ,V,C)
02850C
02860C UPDATE CURRENT ARRAY C BY Y * (DT,DZ) * V.
02870C
02880       COMPLEX S,Y(1),V(1),C(1)
02885       NJ2=NJ/2
02890       DO 100 I=1,NI
02900       S=0.C
02910       IND=I
02920       DO 50 J=1,NJ
02922       D=DT
02924       IF(J.GT.NJ2) D=DZ
02930       S=S+Y(IND)*D*V(J)
02940 50    IND=IND+NI
02950       C(I)=C(I)+S
02960 100   CONTINUE
02970       RETURN
02980       END
03100C ------------------------------------------------
03110       SUBROUTINE MULTRC(NI,DT,DZ,R,C,G)
03120C
03130C UPDATE G BY (DT,DZ) * R * C.
03140C
03150       COMPLEX G,R(1),C(1)
03155       NI2=NI/2
03160       DO 100 I=1,NI
03162       D=DT
03164       IF(I.GT.NI2) D=DZ
03170       G=G+D*R(I)*C(I)
03180 100   CONTINUE
03190       RETURN
03200       END
03210C ************************************************************
03220       FUNCTION SINC(X)
03230C ************************************************************
03240       SINC=1.C
03250       IF(X.NE.0.0) SINC=SIN(X)/X
03260       RETURN
03270       END
```

```
03280C ****************************************************
03290       FUNCTION DBCON(X)
03300C ****************************************************
03310C RETURN X IN DB.
03320       DBCON=-1000.0
03330       IF(X.EQ.0.0) RETURN
03340       DBCON=10.*ALOG10(X)
03350       RETURN
03360       END
03370C ------------------------------------------------------------
03375       SUBROUTINE BOTIN
03380C
03385C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
03390C
03395       COMMON /BOT1/ NMODE,NPT,NBAND
03400       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
03405       COMMON /BOT3/ DH(82),SV(82),CV(82)
03410       COMMON /BOT5/ T(160),TP(160),TZ(160)
03415       COMMON /BOT6/ IEDGE,IUNIF
03420       READ(5,49) NMODE,NPT,NBAND
03425       READ(5,49) NP
03430 49    FORMAT(3I3)
03435       WRITE(6,48) NMODE,NPT,NBAND,NP
03440 48    FORMAT(32H     NMODE    NPT     NBAND      NP,/,4I8)
03445       READ(5,53)(YB(I),I=1,NP)
03450       READ(5,53)(XB(I),I=1,NP)
03455   53  FORMAT(10F8.4)
03460       WRITE(6,55)
03465 55    FORMAT(/,3H YB)
03470       WRITE(6,46)(YB(I),I=1,NP)
03475   46  FORMAT(1X,10F8.4)
03480       WRITE(6,56)
03485 56    FORMAT(/,3H XB)
03490       WRITE(6,46)(XB(I),I=1,NP)
03495C PLOT THE BODY COORDINATES.
03500       CALL PLOTB(XB,YB,NP,41)
03505       READ(5,53) BL
03510       WRITE(6,47) BL
03515 47    FORMAT(//,21H HALF-LENGTH OF BOT =,F12.4)
03520C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
03525       IF(MOD(NP,2).NE.1) GO TO 980
03530C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
03535       IEDGE=1
03540       IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
03545       IEDGE=0
03550       YB(NP+1)=YB(2)
03555       XB(NP+1)=XB(2)
03560       YB(NP+2)=YB(3)
03565       XB(NP+2)=XB(3)
03570       NP=NP+2
03575       WRITE(6,66) NP
03580 66    FORMAT(//,39H BOT GENERATING CURVE IS CLOSED.  NP = ,I3)
```

AD-A118 498 MCDONNELL DOUGLAS RESEARCH LABS ST LOUIS MO F/G 9/2
ALGORITHM FOR SURFACE OF TRANSLATION ATTACHED RADIATORS (A-STAR--ETC(U)
MAY 82 L N MEDGYESI-MITSCHANG, J M PUTNAM F30602-80-C-0106
UNCLASSIFIED RADC-TR-82-113-VOL-3 NL



```
C3585  10    CONTINUE
C3590C COMPUTATION OF BODY SEGMENT PARAMETERS.
C3595        DO 57 I=2,NP
C3600        I2=I-1
C3605        RR1=YB(I)-YB(I2)
C3610        RR2=XB(I)-XB(I2)
C3615        DH(I2)=SQRT(RR1*RR1+RR2*RR2)
C3620        XB1(I2)=.5*(XB(I)+XB(I2))
C3625        YB1(I2)=.5*(YB(I)+YB(I2))
C3630        SV(I2)=RR1/DH(I2)
C3635        CV(I2)=RR2/DH(I2)
C3640  57    CONTINUE
C3645C CHECK IF BOT SEGMENTATION IS UNIFORM.
C3650        IUNIF=0
C3655        NP1=NP-1
C3660        DO 60 I=2,NP1
C3665        RR1=DH(I)/DH(1)
C3670        IF(RR1.LT.0.99 .OR. RR1.GT.1.01) GO TO 20
C3675  60    CONTINUE
C3680        IUNIF=1
C3685        WRITE(6,67)
C3690  67    FORMAT(//,' BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
C3695  20    CONTINUE
C3700C COMPUTATION OF TRIANGLE FUNCTIONS T.
C3705        NM=(NP-3)/2
C3710        DO 74 J=1,NM
C3715        J2=2*(J-1)+1
C3720        J3=J2+1
C3725        J4=J3+1
C3730        J5=J4+1
C3735        J6=4*(J-1)+1
C3740        J7=J6+1
C3745        J8=J7+1
C3750        J9=J8+1
C3755        DEL1=DH(J2)+DH(J3)
C3760        DEL2=DH(J4)+DH(J5)
C3765        TP(J6)=1./DEL1
C3770        TP(J7)=1./DEL1
C3775        TP(J8)=-1./DEL2
C3780        TP(J9)=-1./DEL2
C3785        T(J6)=DH(J2)/2./DEL1
C3790        T(J7)=(DH(J2)+DH(J3)/2.)/DEL1
C3795        T(J8)=(DH(J4)/2.+DH(J5))/DEL2
C3800        T(J9)=DH(J5)/2./DEL2
C3805    74  CONTINUE
C3810        NM4=NM*4
C3815        DO 75 J=1,NM4
C3820  75    TZ(J)=T(J)
C3825        IF(IEDGE.EQ.0) GO TO 76
C3830        TZ(1)=2.-T(1)
C3835        TZ(2)=2.-T(2)
C3840        TZ(NM4-1)=2.-T(NM4-1)
```

```
03845       TZ(NM4)=2.-T(NM4)
03850  76   CONTINUE
03855       RETURN
03860  980  WRITE(6,981)
03865  981  FORMAT(//,' **** EPROR IN BOT INPUT')
03870       STOP
03875       END
04410C ******************************************************
04420       SUBROUTINE PLOTB(X,Y,N,NR)
04430C ******************************************************************
04440C
04450C WRITTEN 2/14/74   BY J. M. PUTNAM        DEPT 220      X23877
04460C
04470C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
04480C     N IS THE NUMBER OF POINTS TO BE PLOTTED.
04490C     NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
04500C     NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
04510C        NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
04520C
04530       REAL X(1),Y(1),HEAD(10)
04540       INTEGER  LINE(101),BLANK,STAR,PLUS
04550       DATA BLANK,STAR,PLUS /1H ,1H*,1H+/
04560       NC=51
04570       N10=(NC-1)/10
04580       WRITE(6,500)
04590  500  FORMAT(//,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
04600       WRITE(6,504)
04610       XMIN=X(1)
04620       XMAX=X(1)
04630       YMIN=Y(1)
04640       YMAX=Y(1)
04650       DO 6 I=1,N
04660       IF(X(I).LT.XMIN) XMIN=X(I)
04670       IF(X(I).GT.XMAX) XMAX=X(I)
04680       IF(Y(I).LT.YMIN) YMIN=Y(I)
04690       IF(Y(I).GT.YMAX) YMAX=Y(I)
04700  6    CONTINUE
04710       DEL=XMAX-XMIN
04720       IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
04730       XMAX=XMIN+DEL
04740       YMAX=YMIN+DEL
04750       DO 5 I =1,N10
04760       Z=I
04770     5 HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
04780       DY=(YMAX-YMIN)/(NR-1)
04790       IEDGE=1
04800       IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
04810       Z=YMAX+DY
04820       YL=Z-DY/2.
04830       DO 7 J=1,NR
04840       DO 8 K=1,NC
04850  8    LINE(K)=BLANK
```

```
04860       Z=Z-DY
04870       YU=YL
04880       YL=Z-DY/2.
04890       DO 9 I=1,N
04900       IF(Y(I).GE.YU) GO TO 9
04910       IF(Y(I).LT.YL) GO TO 9
04920       K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
04930       IF(K.GT.NC) K=NC
04940       LINE(K)=STAR
04950       IF(MOD(I,2).EQ.1) LINE(K)=PLUS
04960       IF(IEDGE.EQ.0) GO TO 9
04970       IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
04980 9     CONTINUE
04990       WRITE(6,508) Z,(LINE(K),K=1,NC)
05000 7     CONTINUE
05010       WRITE(6,504)
05020       WRITE( 6,3002)
05030       WRITE( 6,507) XMIN,(HEAD(I),I=1,N10)
05040       RETURN
05050  504  FORMAT ( 1X, 14(1H-), 1H., 10(5H----.), 1H- )
05060  507  FORMAT(10X,11(F10.4))
05070  508  FORMAT (1X, F12.4,1X, 1HI, 51A1, 1HI )
05080 3002  FORMAT(4X,7HYH / XH,4X,1HI,5(9X,1HI))
05090       END
05100C -------------------------------------------------
05102       SUBROUTINE CAPIN
05104C
05106C READ CAP INPUTS AND COMPUTE CAP ARRAYS.
05108C
05110       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
05112       COMMON /CAP1/ NC,XC,YC,ZC(2)
05114       COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
05116       COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
05118       COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
05120       COMMON /EDG1/ NE,ZE(2),ZBE(10)
05122       COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
05124       READ(5,1) NC,NPR,NE
05126 1     FORMAT(3I3)
05127       IF(NE.NE.0) NE=NC
05128       WRITE(6,3) NC,NPR,NE
05130 3     FORMAT(16H1    NC   NPR    NE,/,3I5,//)
05132       IF(NC.EQ.0) RETURN
05134       READ(5,2) XC,YC
05136 2     FORMAT(10F8.4)
05138       READ(5,2) (ZC(I),I=1,NC)
05142       READ(5,2) (RHOC(I),I=1,NPR)
05143       IF(NE.NE.0) READ(5,2) (ZE(I),I=1,NE)
05144       WRITE(6,4)
05146 4     FORMAT(37H  CAP       XC       YC       ZC       ZE,/)
05148       DO 100 I=1,NC
05149       IF(NE.EQ.0) WRITE(6,5) I,XC,YC,ZC(I)
05150       IF(NE.NE.0) WRITE(6,5) I,XC,YC,ZC(I),ZE(I)
```

```
C5151 100   CONTINUE
C5152 5     FORMAT(I4,4X,4F8.4)
C5154       WRITE(6,6)
C5156 6     FORMAT(//,5H RHOC)
C5158       WRITE(6,7) (RHOC(I),I=1,NPR)
C5160 7     FORMAT(1X,10F8.4)
C5162       IF(MOD(NPR,2).NE.1) GO TO 980
C5164       DO 120 I=2,NPR
C5166 120   IF(RHOC(I).LE.RHOC(I-1)) GO TO 980
C5168C COMPUTATION OF CAP SECTOR PARAMETERS.
C5170       DO 47 I=1,NP
C5172       RC(I)=SQRT((YB(I)-YC)**2+(XB(I)-XC)**2)
C5174 47    CONTINUE
C5176       DO 57 I=2,NP
C5178       I2=I-1
C5180       RR1=YB1(I2)-YC
C5182       RR2=XB1(I2)-XC
C5184       RC1(I2)=SQRT(RR1*RR1+RR2*RR2)
C5186       AC(I2)=ABS((XB(I)-XC)*(YB(I)+YC)+(XB(I2)-XB(I))*
C5188      1  (YB(I2)+YB(I))+(XC-XB(I2))*(YC+YB(I2)))/2.
C5190       SPC(I2)=RR1/RC1(I2)
C5192       CPC(I2)=RR2/RC1(I2)
C5194 57    CONTINUE
C5196       DO 67 I=2,NPR
C5198       I2=I-1
C5200       RHOC1(I2)=(RHOC(I)+RHOC(I2))/2.
C5202       DRHOC(I2)=RHOC(I)-RHOC(I2)
C5204 67    CONTINUE
C5206C COMPUTATION OF CAP TRIANGLE FUNCTIONS.
C5208       LC=(NPR-3)/2
C5210       DO 74 J=1,LC
C5212       J2=2*(J-1)+1
C5214       J3=J2+1
C5216       J4=J3+1
C5218       J5=J4+1
C5220       J6=4*(J-1)+1
C5222       J7=J6+1
C5224       J8=J7+1
C5226       J9=J8+1
C5228       DEL1=DRHOC(J2)+DRHOC(J3)
C5230       DEL2=DRHOC(J4)+DRHOC(J5)
C5232       TPCR(J6)=1./DEL1
C5234       TPCR(J7)=1./DEL1
C5236       TPCR(J8)=-1./DEL2
C5238       TPCR(J9)=-1./DEL2
C5240       TCR(J6)=DRHOC(J2)/2./DEL1
C5242       TCR(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
C5244       TCR(J8)=(DRHOC(J4)/2.+DRHOC(J5))/DEL2
C5246       TCR(J9)=DRHOC(J5)/2./DEL2
C5248   74  CONTINUE
C5250       LC4=LC*4
C5252       DO 75 I=1,LC4
```

```
      TCT(I)=TCR(I)
      TPCT(I)=TPCR(I)
75    CONTINUE
      TCT(LC4-1)=2.-TCT(LC4-1)
      TCT(LC4)=2.-TCT(LC4)
      TPCT(LC4-1)=-TPCT(LC4-1)
      TPCT(LC4)=-TPCT(LC4)
      IF(RHOC(1).EQ.0.0) GO TO 76
      TCT(1)=2.-TCT(1)
      TCT(2)=2.-TCT(2)
      TPCT(1)=-TPCT(1)
      TPCT(2)=-TPCT(2)
76    CONTINUE
      IF(NE.EQ.0) RETURN
C COMPUTATION OF EDGE HALF TRIANGLE FUNCTIONS.
      DO 80 IC=1,NC
      J2=NPR-2
      J3=J2+1
      J6=2*(IC-1)+1
      J7=J6+1
      DEL1=DRHOC(J2)+DRHOC(J3)
      TPCE(J6)=1./DEL1
      TPCE(J7)=1./DEL1
      TCE(J6)=DRHOC(J2)/2./DEL1
      TCE(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
      DEL2=ZE(IC)-ZC(IC)
      ZBE(J6)=ZC(IC)+0.25*DEL2
      ZBE(J7)=ZC(IC)+0.75*DEL2
      IF(DEL2.LT.0.0) GO TO 78
C EDGE IS AT Z=-L.
      TPBE(J6)=-1./DEL2
      TPBE(J7)=-1./DEL2
      TBE(J6)=0.75
      TBE(J7)=0.25
      GO TO 80
78    CONTINUE
C EDGE IS AT Z=+L.
      TPBE(J6)=1./DEL2
      TPBE(J7)=1./DEL2
      TBE(J6)=-0.75
      TBE(J7)=-0.25
80    CONTINUE
      RETURN
980   WRITE(6,981)
981   FORMAT(//,' **** ERROR IN CAP INPUT')
      STOP
      END
C ------------------------------------------------------
      SUBROUTINE WIREIN
C
C READ WIRE COORDINATES AND COMPUTE WIRE SEGMENT ARRAYS.
C
```

```
      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
     1  XW1(100),YW1(100),ZW1(100)
      COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
      COMMON /WIRE3/ NW,INDW(6),RADW(5)
      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
      COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
      NW=0
      NJ=0
      LW=0
      READ(5,51) NW,NPW,NJ
   51 FORMAT(3I3)
      WRITE(6,1) NW,NPW,NJ
    1 FORMAT(24H1      NW      NPW      NJ,/,3I8)
      IF(NW.EQ.0) RETURN
      READ(5,53) (XW(I),I=1,NPW)
      READ(5,53) (YW(I),I=1,NPW)
      READ(5,53) (ZW(I),I=1,NPW)
   53 FORMAT(10F8.4)
      READ(5,52) (INDW(I),I=1,NW)
   52 FORMAT(10I8)
      INDW(NW+1)=NPW+1
      READ(5,53) (RADW(I),I=1,NW)
      IF(NJ.EQ.0) GO TO 50
C NOTE, INDJW MUST BE MONOTONIC INCREASING.
      READ(5,52) (INDJW(I),I=1,NJ)
      READ(5,53) (RADD(I),I=1,NJ)
      READ(5,53) (UXJ(I),I=1,NJ)
      READ(5,53) (UYJ(I),I=1,NJ)
      READ(5,53) (UZJ(I),I=1,NJ)
   50 IERW=0
      IERJ=0
      WRITE(6,61)
   61 FORMAT(/,22X,'WIRE COORDINATES',20X,'JUNCTION PARAMETERS',/,
     1  26X,'IW',4X,'XW',6X,'YW',6X,'ZW',
     2  10X,'IJ',4X,'RADD',5X,'UXJ',5X,'UYJ',5X,'UZJ')
      IJ=1
C THIS LOOP LISTS WIRE/JUNCTION POINTS, WHILE CHECKING THE FOLLOWING:
C   1) EACH WIRE MUST CONTAIN AN ODD NUMBER OF POINTS.
C   2) EACH JUNCTION MUST EITHER START OR TERMINATE A WIRE.
C   3) CHECK THAT ALL JUNCTION POINTS ARE FOUND.
      DO 100 IW=1,NW
      WRITE(6,62) IW,RADW(IW)
   62 FORMAT(2X,'WIRE',I3,' RADW=',F8.4)
      I1=INDW(IW)
      I2=INDW(IW+1)-1
C CHECK FOR AN ODD NUMBER OF POINTS ON WIRE IW.
      IF(MOD(I2-I1+1,2).NE.1) IERW=1
```

```
C5795       DO 90 I=I1,I2
C5800       WRITE(6,63) I,XW(I),YW(I),ZW(I)
C5805 63    FORMAT(25X,I3,3F8.4)
C5810C CHECK IF WIRE POINT I IS A JUNCTION POINT.
C5815       IF(IJ.GT.NJ) GO TO 90
C5820       IF(INDJW(IJ).NE.I) GO TO 90
C5825C CHECK THAT JUNCTION POINT IJ IS AT THE START OR END OF WIRE.
C5830       IF(I.NE.I1 .AND. I.NE.I2) IERJ=1
C5835       WRITE(6,64) IJ,RADD(IJ),UXJ(IJ),UYJ(IJ),UZJ(IJ)
C5840 64    FORMAT('+',58X,I3,4F8.4)
C5845C DETERMINE DIRECTION IN WHICH WIRE LEAVES JUNCTION POINT.
C5850       INDTJ(IJ)=1
C5855       IF(I.EQ.I2) INDTJ(IJ)=-1
C5860C COMPUTE JUNCTION PARAMETERS.
C5865       RADJ(IJ)=RADW(IW)
C5870       XJ(IJ)=XW(I)
C5875       YJ(IJ)=YW(I)
C5880       ZJ(IJ)=ZW(I)
C5885       IJ=IJ+1
C5890 90    CONTINUE
C5895 100   CONTINUE
C5900C CHECK FOR WIRE OR JUNCTION INPUT ERRORS.
C5905       IF(IERW.NE.0) GO TO 980
C5910       IF(IJ-1.NE.NJ .OR. IERJ.NE.0) GO TO 990
C5915C COMPUTATION OF WIRE SEGMENT PARAMETERS.
C5920       DO 57 I=2,NPW
C5925       I2=I-1
C5930       DXW(I2)=XW(I)-XW(I2)
C5935       DYW(I2)=YW(I)-YW(I2)
C5940       DZW(I2)=ZW(I)-ZW(I2)
C5945       DHW(I2)=SQRT(DXW(I2)**2+DYW(I2)**2+DZW(I2)**2)
C5950       XW1(I2)=0.5*(XW(I)+XW(I2))
C5955       YW1(I2)=0.5*(YW(I)+YW(I2))
C5960       ZW1(I2)=0.5*(ZW(I)+ZW(I2))
C5965    57 CONTINUE
C5970C COMPUTATION OF WIRE TRIANGLE FUNCTIONS TW.
C5975       LW=0
C5980       DO 75 IW=1,NW
C5985       I1=INDW(IW)
C5990       I2=INDW(IW+1)-1
C5995       LW1=(I2-I1-2)/2
C6000       DO 74 J=1,LW1
C6005       LW=LW+1
C6010       J2=2*(J-1)+I1
C6015       J3=J2+1
C6020       J4=J3+1
C6025       J5=J4+1
C6030       J6=4*(LW-1)+1
C6035       J7=J6+1
C6040       J8=J7+1
C6045       J9=J8+1
C6050       INDTW(LW)=J2
```

```
C6055         DEL1=DHW(J2)+DHW(J3)
C6060         DEL2=DHW(J4)+DHW(J5)
C6065         TPW(J6)=1./DEL1
C6070         TPW(J7)=1./DEL1
C6075         TPW(J8)=-1./DEL2
C6080         TPW(J9)=-1./DEL2
C6085         TW(J6)=DHW(J2)/2./DEL1
C6090         TW(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C6095         TW(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C6100         TW(J9)=DHW(J5)/2./DEL2
C6105  74     CONTINUE
C6110  75     CONTINUE
C6115C NOTE, LW=(NPW-NW)/2-NW
C6120         IF(NJ.EQ.0) RETURN
C6125C COMPUTATION OF JUNCTION HALF TRIANGLE FUNCTIONS TJ.
C6130         DO 85 IJ=1,NJ
C6135         IF(INDTJ(IJ).GT.0) GO TO 80
C6140C JUNCTION IS AT THE END OF A WIRE.
C6145         J2=INDJW(IJ)-2
C6150         J3=J2+1
C6155         J6=2*(IJ-1)+1
C6160         J7=J6+1
C6165         INDTJ(IJ)=J2
C6170         DEL1=-DHW(J2)-DHW(J3)
C6175         TPJ(J6)=1./DEL1
C6180         TPJ(J7)=1./DEL1
C6185         TJ(J6)=DHW(J2)/2./DEL1
C6190         TJ(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C6195         GO TO 85
C6200C JUNCTION IS AT THE START OF A WIRE.
C6205  80     J4=INDJW(IJ)
C6210         J5=J4+1
C6215         J8=2*(IJ-1)+1
C6220         J9=J8+1
C6225         INDTJ(IJ)=J4
C6230         DEL2=DHW(J4)+DHW(J5)
C6235         TPJ(J8)=-1./DEL2
C6240         TPJ(J9)=-1./DEL2
C6245         TJ(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C6250         TJ(J9)=DHW(J5)/2./DEL2
C6255  85     CONTINUE
C6260C COMPUTE UNIT VECTORS FOR JUNCTION(DISK).
C6265         DO 95 IJ=1,NJ
C6270C NORMAL UNIT VECTOR (UJ).
C6275         RR=SQRT(UXJ(IJ)**2+UYJ(IJ)**2+UZJ(IJ)**2)
C6280         UXJ(IJ)=UXJ(IJ)/RR
C6285         UYJ(IJ)=UYJ(IJ)/RR
C6290         UZJ(IJ)=UZJ(IJ)/RR
C6295C FIND 2 ORTHOGONAL UNIT VECTORS IN THE PLANE OF THE DISK (UJ1 & UJ2).
C6300         UXJ1(IJ)=0.0
C6305         UYJ1(IJ)=0.0
C6310         UZJ1(IJ)=0.0
```

```
06315       UXJ2(IJ)=0.0
06320       UYJ2(IJ)=0.0
06325       UZJ2(IJ)=0.0
06330       IF(UXJ(IJ).EQ.0.0) GO TO 91
06335C FIND INTERSECTION WITH X-Y PLANE.
06340       UYJ1(IJ)=1.0
06345       UXJ1(IJ)=-UYJ(IJ)/UXJ(IJ)
06350C FIND INTERSECTION WITH X-Z PLANE.
06355       UZJ2(IJ)=1.0
06360       UXJ2(IJ)=-UZJ(IJ)/UXJ(IJ)
06365       GO TO 94
06370 91    IF(UYJ(IJ).EQ.0.0) GO TO 92
06375C FIND INTERSECTION WITH Y-Z PLANE.
06380       UZJ1(IJ)=1.0
06385       UYJ1(IJ)=-UZJ(IJ)/UYJ(IJ)
06390C FIND INTERSECTION WITH X-Y PLANE.
06395       UXJ2(IJ)=1.0
06400       UYJ2(IJ)=-UXJ(IJ)/UYJ(IJ)
06405       GO TO 94
06410 92    IF(UZJ(IJ).EQ.0.0) GO TO 94
06415C FIND INTERSECTION WITH X-Z PLANE.
06420       UXJ1(IJ)=1.0
06425       UZJ1(IJ)=-UXJ(IJ)/UZJ(IJ)
06430C FIND INTERSECTION WITH Y-Z PLANE.
06435       UYJ2(IJ)=1.0
06440       UZJ2(IJ)=-UYJ(IJ)/UZJ(IJ)
06445 94    CONTINUE
06450       RR=SQRT(UXJ1(IJ)**2+UYJ1(IJ)**2+UZJ1(IJ)**2)
06455       UXJ1(IJ)=UXJ1(IJ)/RR
06460       UYJ1(IJ)=UYJ1(IJ)/RR
06465       UZJ1(IJ)=UZJ1(IJ)/RR
06470       RR=SQRT(UXJ2(IJ)**2+UYJ2(IJ)**2+UZJ2(IJ)**2)
06475       UXJ2(IJ)=UXJ2(IJ)/RR
06480       UYJ2(IJ)=UYJ2(IJ)/RR
06485       UZJ2(IJ)=UZJ2(IJ)/RR
06490 95    CONTINUE
06495       RETURN
06500 980   WRITE(6,981)
06505 981   FORMAT(//,' **** ERROR IN WIRE INPUT')
06510       STOP
06515 990   WRITE(6,991)
06520 991   FORMAT(//,' **** ERROR IN JUNCTION INPUT')
06525       STOP
06530       END
07940C ------------------------------------------------------
07950       SUBROUTINE ORDER(MORD)
07960C DETERMINE THE ORDER IN WHICH THE Y MATRIX WAS GENERATED, AND
07970C CHECK FOR DIMENSIONAL CONSISTENCY BETWEEN THE Y MATRIX DATA FILE,
07980C AND THE INPUT DATA FILE.
07990       DIMENSION MORD(2)
08000       COMMON /BOT1/ NMODE,NPT,N8AND
08010       COMMON /BOT2/ NP,BL,Y8(83),X8(83),Y81(82),X81(82)
```

```
      COMMON /CAP1/ NC,XC,YC,ZC(2)
      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
     1 XW1(100),YW1(100),ZW1(100)
      COMMON /WIRE3/ NW,INDW(6),RADW(5)
      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
      COMMON /EDG1/ NE,ZE(2),ZBE(10)
      READ(1) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
      IF(NMODE.NE.NMODE1) GO TO 500
      IF(NP.NE.NP1) GO TO 500
      IF(NC.NE.NC1) GO TO 500
      IF(NPR.NE.NPR1) GO TO 500
      IF(NE.NE.NE1) GO TO 500
      IF(NW.NE.NW1) GO TO 500
      IF(NPW.NE.NPW1) GO TO 500
      IF(NJ.NE.NJ1) GO TO 500
C MORD SPECIFIES THE ORDER IN WHICH CAPS AND/OR WIRES HAVE BEEN ADDED
C TO THE BOT.  INDEX 1 REFERS TO CAPS AND INDEX 2 REFERS TO WIRES.
C   MORD(I)=0 IF I HAS NOT BEEN ADDED
C   MORD(I)=N IF I HAS BEEN ADDED TO THE SYSTEM MATRIX, AND
C   IS LOCATED AT PSEUDO MODE NUMBER N.
      READ(1) MORD
      WRITE(6,4) MORD
4     FORMAT(' THE Y MATRIX CONTAINS THE FOLLOWING ADDITIONS ( 0 IF',
     1 ' NOT INCLUDED, OR CORRESPONDING MODE NUMBER IF PRESENT):',//,
     2 10X,'CAPS',I4,/,10X,'WIRES',I3)
      RETURN
500   WRITE(6,1)
1     FORMAT(//,' *** ERROR ***   INPUT PARAMETERS DO NOT CHECK WITH',
     1 ' THE PARAMETERS IN THE Y MATRIX FILE',///,
     2 6X,'  NMODE    NP    NC   NPR    NE    NW   NPW    NJ')
      WRITE(6,2) NMODE,NP,NC,NPR,NE,NW,NPW,NJ
2     FORMAT(' INPUT ',8I6)
      WRITE(6,3) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
3     FORMAT(' Y FILE',8I6)
      STOP
      END
C*************************************************************
      SUBROUTINE RBOT(RBT,RBP,NT,TH,PHI)
C*************************************************************
C COMPUTE BOT TRANSFER MATRICES FOR THETA AND PHI POLARIZATION.
      COMPLEX RBT(1),RBP(1),A6,U
      COMMON /WAVE/ RK
      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
      COMMON /BOT3/ DH(82),SV(82),CV(82)
      COMMON /BOT5/ T(160),TP(160),TZ(160)
      DIMENSION TH(1),PHI(1)
      DATA PI,U /3.1415926,(0.,1.)/
      DATA DTOR /0.017453292/
      LS=NP-3
      NM=LS/2
      DO 156 K=1,NT
```

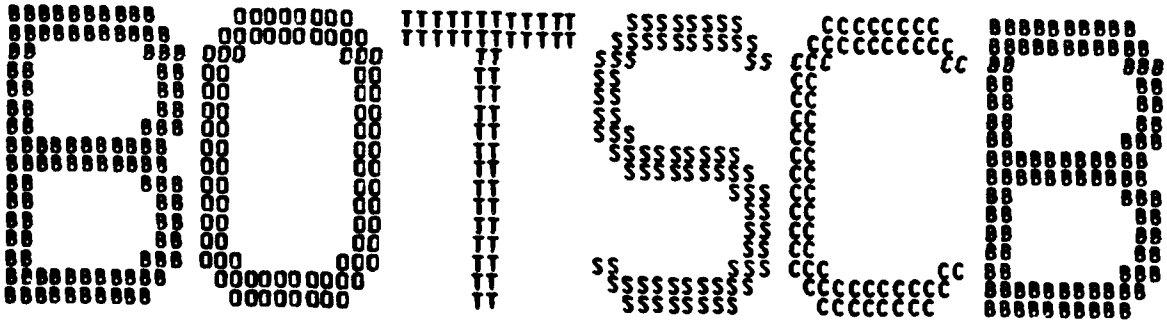
```
      K1=(K-1)*LS
      CS=COS(TH(K)*DTOR)
      SN=SIN(TH(K)*DTOR)
      CSPHI=COS(PHI(K)*DTOR)
      SNPHI=SIN(PHI(K)*DTOR)
      DO 300 J=1,NM
      J1=J+K1
      J2=J1+NM
      RBT(J1)=0.
      RBT(J2)=0.
      RBP(J1)=0.
      RBP(J2)=0.
C     CALCULATION OF TRANSFER MATRICES (EQS. 38-41).
C     NOTE, ONLY THE MODE INDEPENDENT PORTION IS COMPUTED HERE.
      DO 301 I=1,4
      I1=2*(J-1)+I
      I4=4*(J-I)+I
      A6=CEXP(U*BK*SN*(XB1(I1)*CSPHI+YB1(I1)*SNPHI))
C     THETA POLARIZED (EQS. 38-39).
      RBT(J1)=RBT(J1)+DH(I1)*T(I4)*(CV(I1)*CSPHI+SV(I1)*SNPHI)*A6
      RBT(J2)=RBT(J2)+DH(I1)*TZ(I4)*A6
C     PHI  POLARIZED (EQS. 40-41).
      RBP(J1)=RBP(J1)+DH(I1)*T(I4)*(SV(I1)*CSPHI-CV(I1)*SNPHI)*A6
  301 CONTINUE
      RBT(J1)=CS*RBT(J1)
      RBT(J2)=-SN*RBT(J2)
  300 CONTINUE
  156 CONTINUE
      RETURN
      END
C-----------------------------------------------------
      SUBROUTINE RCAP(RCT,RCP,NT,TH,PHI)
C
C     COMPUTE CAP TRANSFER MATRICES FOR THETA AND PHI POLARIZATION.
C
      COMPLEX RCT(1),RCP(1),A6,U
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
      COMMON /BOT3/ DH(82),SV(82),CV(82)
      COMMON /BOT5/ T(160),TP(160),TZ(160)
      COMMON /CAP1/ NC,XC,YC,ZC(2)
      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
      COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
      COMMON /EDG1/ NE,ZE(2),ZBE(10)
      COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
      DIMENSION TH(1),PHI(1)
      DATA PI,U /3.14159265,(0.,1.)/
      DATA DTOR /0.017453292/
      NM=(NP-3)/2
      LR=(NPR-3)/2
      LC=NC*NM*LR
```

```
      LE=NE*NM
      DO 156 K=1,NT
      K1=(K-1)*(2*LC+LE)
      CS=COS(TH(K)*DTOR)
      SN=SIN(TH(K)*DTOR)
      CSPHI=COS(PHI(K)*DTOR)
      SNPHI=SIN(PHI(K)*DTOR)
      J1=K1
      DO 300 JC=1,NC
      DO 300 JP=1,NM
      DO 300 JR=1,LR
      J1=J1+1
      J2=J1+LC
      RCT(J1)=0.
      RCT(J2)=0.
      RCP(J1)=0.
      RCP(J2)=0.
C CALCULATION OF TRANSFER MATRICES.
      DO 301 IIP=1,4
      IP1=2*(JP-1)+IIP
      IP4=4*(JP-1)+IIP
      DO 301 IIR=1,4
      IR1=2*(JR-1)+IIR
      IR4=4*(JR-1)+IIR
      XX=XC+RHOC1(IR1)*RC1(IP1)*CPC(IP1)
      YY=YC+RHOC1(IR1)*RC1(IP1)*SPC(IP1)
      ZZ=ZC(JC)
      A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
      AI=AC(IP1)*ABS(RHOC(IR1+1)**2-RHOC(IR1)**2)
C THETA POLARIZED.
      RCT(J1)=RCT(J1)+AI*(CV(IP1)*CSPHI+SV(IP1)*SNPHI)*
     1 T(IP4)*TCT(IR4)*A6
      RCT(J2)=RCT(J2)+AI*(CPC(IP1)*CSPHI+SPC(IP1)*SNPHI)*
     1 TZ(IP4)*TCR(IR4)*A6
C PHI POLARIZED.
      RCP(J1)=RCP(J1)+AI*(SV(IP1)*CSPHI-CV(IP1)*SNPHI)*
     1 T(IP4)*TCT(IR4)*A6
      RCP(J2)=RCP(J2)+AI*(SPC(IP1)*CSPHI-CPC(IP1)*SNPHI)*
     1 TZ(IP4)*TCR(IR4)*A6
  301 CONTINUE
      RCT(J1)=CS*RCT(J1)
      RCT(J2)=CS*RCT(J2)
  300 CONTINUE
      IF(NE.EQ.0) GO TO 156
C EDGE TRANSFER MATRICES FOLLOW.
      J2=2*LC+K1
      DO 400 JC=1,NC
      DO 400 JP=1,NM
      J2=J2+1
      RCT(J2)=0.
      RCP(J2)=0.
      DO 401 IIP=1,4
```

```
09185       IP1=2*(JP-1)+IIP
09186       IP4=4*(JP-1)+IIP
09187       DO 401 IIRZ=1,2
09188       IR1=2*LR+IIRZ
09189       IR4=2*(JC-1)+IIRZ
09190       IZ1=2*(JC-1)+IIRZ
09191       IZ4=2*(JC-1)+IIRZ
09192       XX=XC+RHOC1(IR1)*RC1(IP1)*CPC(IP1)
09193       YY=YC+RHOC1(IR1)*RC1(IP1)*SPC(IP1)
09194       ZZ=ZC(JC)
09195       A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
09196       AI=AC(IP1)*ABS(RHOC(IR1+1)**2-RHOC(IR1)**2)
09197C EDGE(CAP)
09198C THETA POLARIZED.
09199       RCT(J2)=RCT(J2)+CS*AI*(CPC(IP1)*CSPHI+SPC(IP1)*SNPHI)*
09200      1 TZ(IP4)*TCE(IR4)*A6
09201C PHI POLARIZED.
09202C     RCP(J2)=RCP(J2)+AI*(SPC(IP1)*CSPHI-CPC(IP1)*SNPHI)*
09203      1 TZ(IP4)*TCE(IR4)*A6
09204       XX=XB1(IP1)
09205       YY=YB1(IP1)
09206       ZZ=ZBE(IZ1)
09207       A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
09208       AI=DH(IP1)*ABS(ZE(JC)-ZC(JC))/2.
09209C EDGE(BOT).
09210C THETA POLARIZED.
09211       RCT(J2)=RCT(J2)-SN*AI*TZ(IP4)*TBE(IZ4)*A6
09212  401  CONTINUE
09213  400  CONTINUE
09214  156  CONTINUE
09215       RETURN
09216       END
09222C -------------------------------------------------------
09223       SUBROUTINE RWIRE(RWT,RWP,NT,TH,PHI)
09224C
09225C COMPUTE WIRE-JUNCTION TRANSFER MATRICES.
09230C
09240       COMPLEX RWT(1),RWP(1),A6,U
09245       COMMON /WAVE/ BK
09250       COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
09260      1 XW1(100),YW1(100),ZW1(100)
09270       COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
09280       COMMON /WIRE3/ NW,INDW(6),RADW(5)
09290       COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
09300       COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
09310       COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
09320       COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
09325       COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
09326       COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
09340       DIMENSION TH(1),PHI(1)
09350       DATA PI,U /3.14159265,(0.,1.)/
09360       DATA DTOR /0.0174533292/
```

```
09380       LWJ=LW+NJ
09390       DO 156 K=1,NT
09400       K1=(K-1)*LWJ
09410       CS=COS(TH(K)*DTOR)
09420       SN=SIN(TH(K)*DTOR)
09430       CSPHI=COS(PHI(K)*DTOR)
09440       SNPHI=SIN(PHI(K)*DTOR)
09450       DO 300 J=1,LW
09455       J1=J+K1
09460       RWT(J1)=0.
09465       RWP(J1)=0.
09470       DO 301 I=1,4
09475       I1=INDTW(J)+I-1
09480       I4=4*(J-1)+I
09485       A6=CEXP(U*BK*(SN*(XW1(I1)*CSPHI+YW1(I1)*SNPHI)+ZW1(I1)*CS))
09490       RWT(J1)=RWT(J1)+(CS*(DXW(I1)*CSPHI+DYW(I1)*SNPHI)-
09495      1  DZW(I1)*SN)*TW(I4)*A6
09500       RWP(J1)=RWP(J1)+(-DXW(I1)*SNPHI+DYW(I1)*CSPHI)*TW(I4)*A6
09505  301  CONTINUE
09510  300  CONTINUE
09515       IF(NJ.EQ.0) GO TO 156
09520       J1=LW+K1
09525       DO 400 J=1,NJ
09530       J1=J1+1
09535       RWT(J1)=0.
09540       RWP(J1)=0.
09545       DO 401 I=1,2
09550       I1=INDTJ(J)+I-1
09555       I2=2*(J-1)+I
09560       A6=CEXP(U*BK*(SN*(XW1(I1)*CSPHI+YW1(I1)*SNPHI)+ZW1(I1)*CS))
09565       RWT(J1)=RWT(J1)+(CS*(DXW(I1)*CSPHI+DYW(I1)*SNPHI)-
09570      1  DZW(I1)*SN)*TJ(I2)*A6
09575       RWP(J1)=RWP(J1)+(-DXW(I1)*SNPHI+DYW(I1)*CSPHI)*TJ(I2)*A6
09580  401  CONTINUE
09585       DO 402 JJ=1,4
09590       ALPHA=PI*((JJ-1)/2.+0.25)
09595       SA=SIN(ALPHA)
09600       CA=COS(ALPHA)
09605       RA=(RADJ(J)+RADD(J))/2.
09610       XA=XJ(J)+RA*(CA*UXJ1(J)+SA*UXJ2(J))
09612       YA=YJ(J)+RA*(CA*UYJ1(J)+SA*UYJ2(J))
09614       ZA=ZJ(J)+RA*(CA*UZJ1(J)+SA*UZJ2(J))
09616       DT=CA*(UXJ1(J)*CS*CSPHI+UYJ1(J)*CS*SNPHI-UZJ1(J)*SN)+
09618      1  SA*(UXJ2(J)*CS*CSPHI+UYJ2(J)*CS*SNPHI-UZJ2(J)*SN)
09620       DP=CA*(-UXJ1(J)*SNPHI+UYJ1(J)*CSPHI)+
09622      1  SA*(-UXJ2(J)*SNPHI+UYJ2(J)*CSPHI)
09625       RR=(RADJ(J)-RADD(J))/2.
09630       A6=CEXP(U*BK*(SN*(XA*CSPHI+YA*SNPHI)+ZA*CS))
09635       RWT(J1)=RWT(J1)+RR/4.*DT*A6
09645       RWP(J1)=RWP(J1)+RR/4.*DP*A6
09650  402  CONTINUE
09660  400  CONTINUE
```

```
09670   156 CONTINUE
09680       RETURN
09690       END
09700/EOR
```



```
00100BOTS,P30,T100,CM14C000,STCX3.
00110ACCOUNT,M03265,BOT2A.
00120BANNERS(OUTPUT) *J. PUTNAM*DEPT 220*BLD 110-4**
00130ATTACH(INFIL=YSSX)
00140FTN(R=0,OPT=1)
00164LDSET(PRESET=INDEF)
00170LGO.
00190/EOR
00200       PROGRAM BOTSCB(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
00210      1 INFIL,TAPE1=INFIL)
00220C
00230C BOTSCB IS THE BOT BI-STATIC SCATTERING CODE.
00240C
00250C UNIT 5 IS THE CARD READER.
00260C UNIT 6 IS THE LINE PRINTER.
00270C UNIT 1 IS A DISK FILE CONTAINING THE Y MATRIX.
00280C
00285       COMPLEX U
00300       COMPLEX STT(91),SPP(91),STP(91),SPT(91)
00305       COMPLEX CBT(1680),CBP(1680),CWT(59),CWP(59),CCT(96),CCP(96)
00310       COMPLEX ESCT(3),ESCP(3),HSCT(3),HSCP(3)
00320       COMPLEX Y(2304)
00325       COMPLEX RBT(82),RBP(82),RWT(59),RWP(59),RCT(96),RCP(96)
00327       COMMON /WAVE/ BK
00330       COMMON /BOT1/ NMODE,NPT,NBAND
00340       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
00350       COMMON /BOT3/ D4(82),SV(82),CV(82)
00370       COMMON /BOT5/ T(160),TP(160),TZ(160)
00380       COMMON /BOT6/ IEDGE,IUNIF
00382       COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
00384      1 XW1(100),YW1(100),ZW1(100)
00385       COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
00386       COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
00387       COMMON /CAP1/ NC,XC,YC,ZC(2)
00388       COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
00389       COMMON /EDG1/ NE,ZE(2),ZBE(10)
00390       COMMON /INT/ M,RHO2,ZP
00400       DIMENSION THS(73),PHIS(73)
00410       DIMENSION ANG(6),IPLANE(6),ANG1(6),ANG2(6)
00420       DIMENSION NORD(2)
00430       U=(0.,1.)
00440       ETA=376.707
00450       PI=3.14159265
00460       DTOR=PI/180.
00462       READ(5,1) BK
00463  1    FORMAT(E15.7)
00464       WRITE(6,2) BK
00465  2    FORMAT(9H1       BK=,/,E15.7)
00470C
00480       CALL BOTIN
00490C
00500       CALL CAPIN
```

```
00510C
C0520       CALL WIREIN
00522C
C0530C READ INPUTS, DESCRIBING THE INCIDENT FIELD AND SCATTERING PLANES.
00540       READ(5,44) NANG,NT,PHII,THI
00550 44    FORMAT(2I3,2F8.4)
00590       READ(5,53) (ANG(I),I=1,NANG)
C0600 53    FORMAT(10F8.4)
00610C IPLANE = 1 FOR PHI FIXED.
C0620C          2 FOR THETA FIXED.
00630       READ(5,54) (IPLANE(I),I=1,NANG)
00640 54    FORMAT(10I8)
00642       READ(5,53) (ANG1(I),I=1,NANG)
00644       READ(5,53) (ANG2(I),I=1,NANG)
00650       WRITE(6,3) NANG,NT
00660 3     FORMAT(//,25H NUMBER OF FIXED ANGLES =,I3,//
00670      1 35H NUMBER OF ANGLES PER FIXED ANGLE =,I4)
00680       WRITE(6,4) (ANG(I),IPLANE(I),ANG1(I),ANG2(I),I=1,NANG)
00690 4     FORMAT(//,45H FIXED ANGLE    CODE     VARIABLE ANGLE RANGE,
C0700      1 //,(F8.1,9X,I2,6X,F8.1,3H - ,F8.1))
00710       LS=NP-3
C0712       NM=LS/2
00714       NM1=NM+1
00716       NM4=NM*4
00720       LSS=LS*LS
00722       LR=(NPR-3)/2
00724       LC=NC*NM*LR
00726       LC2=LC*2
00728       LE=NE*NM
00730       LSC=LS*(LC2+LE)
00735       LWJ=LW+NJ
00740       LSW=LS*LWJ
00750       LCC=(LC2+LE)**2
00760       LCW=(LC2+LE)*LWJ
00770       LWW=LWJ**2
00810       BKL=BK*BL
C0825       P1=(BK*BK*ETA/2./PI)**2/4./PI
00830       KMODE=2*NMODE-1
00840       CALL ZERO(LS*KMODE,CBT)
00845       CALL ZERO(LS*KMODE,CBP)
00860       IF(NC.NE.0) CALL ZERO(LC2+LE,CCT)
00865       IF(NC.NE.0) CALL ZERO(LC2+LE,CCP)
C0870       IF(NPW.NE.0) CALL ZERO(LWJ,CWT)
00875       IF(NPW.NE.0) CALL ZERO(LWJ,CWP)
00877       IF(NPW.NE.0) CALL ZERO(LWJ,CW)
00880       CALL ORDER(MORD)
00890C CALCULATE CURRENTS, EQ. 10.
00892       CALL RBOT(RBT,RBP,1,THI,PHII)
00894       IF(NC.NE.0) CALL RCAP(RCT,RCP,1,THI,PHII)
00896       IF(NPW.NE.0) CALL RWIRE(RWT,RWP,1,THI,PHII)
00900       NUMB=0
00910 50    CONTINUE
```

```
C0920         READ(1) M,N
C0930         IF(ECF(1)) 37,60
C0940  60     NUMB=NUMB+1
C0942         IC=(N+NMODE-1)*LS+1
C0950C DETERMINE SUBMATRIX TYPE, AND CALCULATE CURRENTS.
C0960         IF(N.LT.NMODE) GO TO 61
C0970         IF(N.EQ.MORD(1)) GO TO 62
C0980         IF(N.EQ.MORD(2)) GO TO 63
C0990         GO TO 50
C0992  61     DNT=2.*BL*SINC(-N*PI+BKL*COS(THI*DTOR))
C0994         DNZ=DNT-2.*BL*(-1)**N*SINC(BKL*COS(THI*DTOR))
C1010         IF(M.LT.NMODE) GO TO 71
C1020         IF(M.EQ.MORD(1)) GO TO 72
C1030         IF(M.EQ.MORD(2)) GO TO 73
C1040         GO TO 50
C1050  62     IF(M.LT.NMODE) GO TO 74
C1060         IF(M.EQ.MORD(1)) GO TO 75
C1070         IF(M.EQ.MORD(2)) GO TO 76
C1080         GO TO 50
C1090  63     IF(M.LT.NMODE) GO TO 77
C1100         IF(M.EQ.MORD(1)) GO TO 78
C1110         IF(M.EQ.MORD(2)) GO TO 79
C1120         GO TO 50
C1130C BOT-BOT.
C1140  71     READ(1) (Y(I),I=1,LSS)
C1170         CALL MULTYR(LS,LS,Y,DNT,DNZ,RBT,CBT(IC))
C1175         CALL MULTYR(LS,LS,Y,DNT,DNZ,RBP,CBP(IC))
C1180         GO TO 50
C1190C CAP-BOT.
C1200  72     READ(1) (Y(I),I=1,LSC)
C1210         CALL MULTYR(LC2+LE,LS,Y,DNT,DNZ,RBT,CCT)
C1215         CALL MULTYR(LC2+LE,LS,Y,DNT,DNZ,RBP,CCP)
C1220         GO TO 50
C1230C WIRE-BOT.
C1240  73     READ(1) (Y(I),I=1,LSW)
C1250         CALL MULTYR(LWJ,LS,Y,DNT,DNZ,RBT,CWT)
C1255         CALL MULTYR(LWJ,LS,Y,DNT,DNZ,RBP,CWP)
C1260         GO TO 50
C1270C BOT-CAP.
C1280  74     READ(1) (Y(I),I=1,LSC)
C1290         CALL MULTYR(LS,LC2+LE,Y,1.0,1.0,RCT,CBT(IC))
C1295         CALL MULTYR(LS,LC2+LE,Y,1.0,1.0,RCP,CBP(IC))
C1300         GO TO 50
C1310C CAP-CAP.
C1320  75     READ(1) (Y(I),I=1,LCC)
C1330         CALL MULTYR(LC2+LE,LC2+LE,Y,1.0,1.0,RCT,CCT)
C1335         CALL MULTYR(LC2+LE,LC2+LE,Y,1.0,1.0,RCP,CCP)
C1340         GO TO 50
C1350C WIRE-CAP.
C1360  76     READ(1) (Y(I),I=1,LCW)
C1370         CALL MULTYR(LWJ,LC2+LE,Y,1.0,1.0,RCT,CWT)
C1375         CALL MULTYR(LWJ,LC2+LE,Y,1.0,1.0,RCP,CWP)
```

```
01380       GO TO 50
01390C BOT-WIRE.
01400  77   READ(1) (Y(I),I=1,LSW)
01410       CALL MULTYR(LS,LWJ,Y,1.0,1.0,RWT,CBT(IC))
01415       CALL MULTYR(LS,LWJ,Y,1.0,1.0,RWP,CBP(IC))
01420       GO TO 50
01430C CAP-WIRE.
01440  78   READ(1) (Y(I),I=1,LCW)
01450       CALL MULTYR(LC2+LE,LWJ,Y,1.0,1.0,RWT,CCT)
01455       CALL MULTYR(LC2+LE,LWJ,Y,1.0,1.0,RWP,CCP)
01460       GO TO 50
01470C WIRE-WIRE.
01480  79   READ(1) (Y(I),I=1,LWW)
01490       CALL MULTYR(LWJ,LW+NJ,Y,1.0,1.0,RWT,CWT)
01495       CALL MULTYR(LWJ,LW+NJ,Y,1.0,1.0,RWP,CWP)
01500       GO TO 50
01550C ALL Y SUBMATRICES HAVE BEEN READ.
01560  37   CONTINUE
01562       WRITE(6,114) NUMB
01564 114   FORMAT(///,1X,I4,17H SUBMATRICES READ)
01630C *** END CALCULATION OF CURRENTS ***
01650       DO 90 IANG=1,NANG
01660C INITIALIZE ARRAYS, AND SET THE SCATTERING ANGLES.
01670       IPLAN=IPLANE(IANG)
01675       DT=(ANG2(IANG)-ANG1(IANG))/(NT-1)
01680       DO 100 K=1,NT
01685       STTR(K)=0.0
01690       SPP(K)=0.0
01695       STP(K)=0.0
01700       SPT(K)=0.0
01710       PHIS(K)=ANG(IANG)*(2-IPLAN)+(ANG1(IANG)+DT*(K-1))*(IPLAN-1)
01720 100   THS(K)=ANG(IANG)*(IPLAN-1)+(ANG1(IANG)+DT*(K-1))*(2-IPLAN)
01725       WRITE(6,403) PHII,THI
01730 403   FORMAT(1H1,22X,34HBI-STATIC RCS(DB).  INCIDENT PHI =,F6.1,
01735      1 9H  THETA =,F6.1,//,20X,45(1H-),/,
01740      2 55H      PHI      THETA          00           60           00
01745      3 8H      00,/,
01750      4 55H+                            --           //           -/
01755      5 8H      /-,/)
01770C CALCULATE RCS.
01780       DO 110 K=1,NT
01790       CALL RBGT(RBT,RBP,1,THS(K),PHIS(K))
01800       IC=0
01810       DO 115 MM=1,KMODE
01820       M=-NMODE+MM
01830C CALCULATE DM AS A FUNCTION OF THETA (EQ. 41).
01840       DMT=2.*BL*SINC(M*PI+BKL*COS(THS(K)*DTOR))
01850       DMZ=DMT-2.*BL*(-1)**M*SINC(BKL*COS(THS(K)*DTOR))
01860       DO 105 J=1,LS
01870       IC=IC+1
01880       DM=DMT
01890       IF(J.GT.NM) DM=DMZ
```

```
01895        STT(K)=STT(K)+DM*RBT(J)*CBT(IC)
01900        SPP(K)=SPP(K)+DM*RBP(J)*CBP(IC)
01905        STP(K)=STP(K)+DM*RBT(J)*CBP(IC)
01910  105   SPT(K)=SPT(K)+DM*RBP(J)*CBT(IC)
01920  115   CONTINUE
01921        IF(LC.EQ.0) GO TO 106
01922        CALL RCAP(RCT,RCP,1,THS(K),PHIS(K))
01924        CALL MULTRC(LC2+LE,RCT,CCT,STT(K))
01925        CALL MULTRC(LC2+LE,RCP,CCP,SPP(K))
01926        CALL MULTRC(LC2+LE,RCT,CCP,STP(K))
01927        CALL MULTRC(LC2+LE,RCP,CCT,SPT(K))
01928  106   CONTINUE
01930        IF(NPW.EQ.0) GO TO 107
01932        CALL RWIRE(RWT,RWP,1,THS(K),PHIS(K))
01934        CALL MULTRC(LWJ,RWT,CWT,STT(K))
01935        CALL MULTRC(LWJ,RWP,CWP,SPP(K))
01936        CALL MULTRC(LWJ,RWT,CWP,STP(K))
01937        CALL MULTRC(LWJ,RWP,CWT,SPT(K))
01938  107   CONTINUE
01939        X2=CABS(STT(K))
01940        X2=DBCON(P1*X2*X2)
01945        X3=CABS(SPP(K))
01950        X3=DBCON(P1*X3*X3)
01955        X4=CABS(STP(K))
01960        X4=DBCON(P1*X4*X4)
01965        X5=CABS(SPT(K))
01970        X5=DBCON(P1*X5*X5)
01975        WRITE(6,113) PHIS(K),THS(K),X2,X3,X4,X5
01980  113   FORMAT(1X,2F8.1,4F12.2)
01990  110   CONTINUE
02000  90    CONTINUE
02010C  *** END CALCULATION OF RCS ***
02020C LIST AND PLOT CURRENTS.
02050        WRITE(6,301)
02055  301   FORMAT(32H1 THETA POLARIZED INCIDENT FIELD,//)
02060        CALL LPCUR(NMODE,NM,CBT,LW,NJ,CWT,LC,LE,CCT)
02065        WRITE(6,302)
02070  302   FORMAT(30H1 PHI POLARIZED INCIDENT FIELD,//)
02075        CALL LPCUR(NMODE,NM,CBP,LW,NJ,CWP,LC,LE,CCP)
02080C  **** END LISTING AND PLOTTING OF CURRENTS ****
02215C
02220C  NEAR FIELD ANALYSIS FOLLOWS.
02225C
02230C  NOTE, ESC IS FOR ELECTRIC FIELDS
02235C        HSC IS FOR MAGNETIC FIELDS
02240C        INDICES 1,2,AND 3 ARE FOR THE X,Y,AND Z COMPONENTS OF THE
02245C        FIELDS, RESPECTIVELY.
02250C        THE T ENDING IS FOR THETA INCIDENT POLARIZATION
02255C        THE P ENDING IS FOR PHI INCIDENT POLARIZATION.
02260C
02265  400   READ(5,44) NTEST
02270        IF(NTEST.EQ.0) GO TO 900
```

```
      WRITE(6,404)
  404 FORMAT(1H1,//,21H  NEAR FIELD ANALYSIS,//
     1 59X,7HE-FIELD,37X,7HH-FIELD,/,
     2 43X,39(1H-),5X,39(1H-),/,
     3 35X,16HFIELD COMPONENTS,28X,16HFIELD COMPONENTS,///,
     4 24H    ZTEST    YTEST    XTEST,
     5 11X,2(14X,6H   X   ,7X,5H  Y  ,9X,3H Z ),/)
      DO 490 ITEST=1,NTEST
      READ(5,53) ZTEST,YTEST,XTEST
      ESCT(1)=0.
      ESCT(2)=0.
      ESCT(3)=0.
      HSCT(1)=0.
      HSCT(2)=0.
      HSCT(3)=0.
      ESCP(1)=0.
      ESCP(2)=0.
      ESCP(3)=0.
      HSCP(1)=0.
      HSCP(2)=0.
      HSCP(3)=0.
C BOT CONTRIBUTION.
C STEP THROUGH MODES.
      M=-NMODE
      DO 430 MM=1,KMODE
      M=M+1
C CALCULATE M-TH MODAL CURRENT COEFFICIENTS (EQS. 67 & 71).
      CALL NEARB(XTEST,YTEST,ZTEST,Y)
C CALCULATE AND SUM Y*CUR, USING THE FIRST 6 ROWS OF Y, YIELDING
C THE 3 FIELD COMPONENTS FOR BOTH ELECTRIC AND MAGNETIC NEAR FIELDS,
C FOR BOTH INCIDENT POLARIZATIONS, (EQS. 67 & 71).
      DO 420 K=1,3
      I1=(M+NMODE-1)*LS
      J1=K-6
      DO 420 I=1,LS
      I1=I1+1
      J1=J1+6
      J2=J1+3
      ESCT(K)=ESCT(K)+Y(J1)*CBT(I1)
      HSCT(K)=HSCT(K)+Y(J2)*CBT(I1)
      ESCP(K)=ESCP(K)+Y(J1)*CBP(I1)
      HSCP(K)=HSCP(K)+Y(J2)*CBP(I1)
  420 CONTINUE
  430 CONTINUE
      IF(NC.EQ.0) GO TO 450
C CAP CONTRIBUTION.
      CALL NEARC(XTEST,YTEST,ZTEST,Y)
      DO 440 K=1,3
      J1=K-6
      DO 440 I=1,LC2
      J1=J1+6
      J2=J1+3
```

```
C2508       ESCT(K)=ESCT(K)+Y(J1)*CCT(I)
C2509       HSCT(K)=HSCT(K)+Y(J2)*CCT(I)
C2510       ESCP(K)=ESCP(K)+Y(J1)*CCP(I)
C2511       HSCP(K)=HSCP(K)+Y(J2)*CCP(I)
C2512  440  CONTINUE
C2514  450  IF(NPW.EQ.0) GO TO 470
C2516C  WIRE  CONTRIBUTION.
C2518       CALL NEARW(XTEST,YTEST,ZTEST,Y)
C2520       DO 460 K=1,3
C2522       J1=K-6
C2524       DO 460 I=1,LW
C2526       J1=J1+6
C2528       J2=J1+3
C2530       ESCT(K)=ESCT(K)+Y(J1)*CWT(I)
C2531       HSCT(K)=HSCT(K)+Y(J2)*CWT(I)
C2532       ESCP(K)=ESCP(K)+Y(J1)*CWP(I)
C2533       HSCP(K)=HSCP(K)+Y(J2)*CWP(I)
C2534  460  CONTINUE
C2536  470  WRITE(6,505) ZTEST,YTEST,XTEST
C2537  505  FORMAT(/,1X,3F8.4)
C2565       WRITE(6,506) (CABS(ESCT(I)),I=1,3),(CABS(HSCT(I)),I=1,3)
C2570  506  FORMAT(26X,17H0 INC POL E-FIELD,3E13.4,5X,3E13.4,/,1H+,25X,1H-)
C2610       WRITE(6,507) (CABS(ESCP(I)),I=1,3),(CABS(HSCP(I)),I=1,3)
C2615  507  FORMAT(26X,17H0 INC POL E-FIELD,3E13.4,5X,3E13.4,/,1H+,25X,1H/)
C2620  490  CONTINUE
C2625  900  STOP
C2645       END
C2821C ------------------------------------------------------
C2822       SUBROUTINE ZERO(N,A)
C2823C A=0.
C2824       COMPLEX A(1)
C2825       DO 10C I=1,N
C2826       A(I)=0.0
C2827  10C  CONTINUE
C2828       RETURN
C2829       END
C2830C ------------------------------------------------------------
C2840       SUBROUTINE MULTYR(NI,NJ,Y,DT,DZ,V,C)
C2850C
C2860C  UPDATE CURRENT ARRAY C BY Y * (DT,DZ) * V.
C2870C
C2880       COMPLEX S,Y(1),V(1),C(1)
C2885       NJ2=NJ/2
C2890       DO 100 I=1,NI
C2900       S=0.0
C2910       IND=I
C2920       DO 5C J=1,NJ
C2922       D=DT
C2924       IF(J.GT.NJ2) D=DZ
C2930       S=S+Y(IND)*D*V(J)
C2940  5C   IND=IND+NI
C2950       C(I)=C(I)+S
```

```
 02960  100  CONTINUE
 02970       RETURN
 02980       END
 03100C ------------------------------------------------------
 03110       SUBROUTINE MULTRC(NI,R,C,G)
 03120C
 03130C UPDATE G BY R * C.
 03140C
 03150       COMPLEX G,R(1),C(1)
 03160       DO 100 I=1,NI
 03170       G=G+R(I)*C(I)
 03180  100  CONTINUE
 03190       RETURN
 03200       END
 03210C ************************************************************
 03220       FUNCTION SINC(X)
 03230C ************************************************************
 03240       SINC=1.0
 03250       IF(X.NE.0.0) SINC=SIN(X)/X
 03260       RETURN
 03270       END
 03280C ***********************************************
 03290       FUNCTION DBCON(X)
 03300C ***********************************************
 03310C RETURN X IN DB.
 03320       DBCON=-1000.0
 03330       IF(X.EQ.0.0) RETURN
 03340       DBCON=10.*ALOG10(X)
 03350       RETURN
 03360       END
 03370C ------------------------------------------------------
 03375       SUBROUTINE BOTIN
 03380C
 03385C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
 03390C
 03395       COMMON /BOT1/ NMODE,NPT,NBAND
 03400       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
 03405       COMMON /BOT3/ DH(82),SV(82),CV(82)
 03410       COMMON /BOT5/ T(160),TP(160),TZ(160)
 03415       COMMON /BOT6/ IEDGE,IUNIF
 03420       READ(5,49) NMODE,NPT,NBAND
 03425       READ(5,49) NP
 03430  49   FORMAT(3I3)
 03435       WRITE(6,48) NMODE,NPT,NBAND,NP
 03440  48   FORMAT(32H     NMODE    NPT     NBAND      NP,/,4I8)
 03445       READ(5,53)(YB(I),I=1,NP)
 03450       READ(5,53)(XB(I),I=1,NP)
 03455    53 FORMAT(10F8.4)
 03460       WRITE(6,55)
 03465  55   FORMAT(/,3H YB)
 03470       WRITE(6,46)(YB(I),I=1,NP)
 03475    46 FORMAT(1X,10F8.4)
```

```
C3480       WRITE(6,56)
C3485  56    FORMAT(/,3H XB)
C3490       WRITE(6,46)(XB(I),I=1,NP)
C3495C PLOT THE BODY COORDINATES.
C3500       CALL PLOTB(XB,YB,NP,41)
C3505       READ(5,53) BL
C3510       WRITE(6,47) BL
C3515  47    FORMAT(//,21H HALF-LENGTH OF BOT =,F12.4)
C3520C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
C3525       IF(MOD(NP,2).NE.1) GO TO 980
C3530C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
C3535       IEDGE=1
C3540       IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
C3545       IEDGE=0
C3550       YB(NP+1)=YB(2)
C3555       XB(NP+1)=XB(2)
C3560       YB(NP+2)=YB(3)
C3565       XB(NP+2)=XB(3)
C3570       NP=NP+2
C3575       WRITE(6,66) NP
C3580  66    FORMAT(//,39H BOT GENERATING CURVE IS CLOSED.  NP = ,I3)
C3585  10    CONTINUE
C3590C COMPUTATION OF BODY SEGMENT PARAMETERS.
C3595       DO 57 I=2,NP
C3600       I2=I-1
C3605       RR1=YB(I)-YB(I2)
C3610       RR2=XB(I)-XB(I2)
C3615       DH(I2)=SQRT(RR1*RR1+RR2*RR2)
C3620       XB1(I2)=.5*(XB(I)+XB(I2))
C3625       YB1(I2)=.5*(YB(I)+YB(I2))
C3630       SV(I2)=RR1/DH(I2)
C3635       CV(I2)=RR2/DH(I2)
C3640  57    CONTINUE
C3645C CHECK IF BOT SEGMENTATION IS UNIFORM.
C3650       IUNIF=0
C3655       NP1=NP-1
C3660       DO 60 I=2,NP1
C3665       RR1=DH(I)/DH(1)
C3670       IF(RR1.LT.0.99 .OR. RR1.GT.1.01) GO TO 20
C3675  60    CONTINUE
C3680       IUNIF=1
C3685       WRITE(6,67)
C3690  67    FORMAT(//,' BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
C3695  20    CONTINUE
C3700C COMPUTATION OF TRIANGLE FUNCTIONS T.
C3705       NM=(NP-3)/2
C3710       DO 74 J=1,NM
C3715       J2=2*(J-1)+1
C3720       J3=J2+1
C3725       J4=J3+1
C3730       J5=J4+1
C3735       J6=4*(J-1)+1
```

```
03740       J7=J6+1
03745       J8=J7+1
03750       J9=J8+1
03755       DEL1=DH(J2)+DH(J3)
03760       DEL2=DH(J4)+DH(J5)
03765       TP(J6)=1./DEL1
03770       TP(J7)=1./DEL1
03775       TP(J8)=-1./DEL2
03780       TP(J9)=-1./DEL2
03785       T(J6)=DH(J2)/2./DEL1
03790       T(J7)=(DH(J2)+DH(J3)/2.)/DEL1
03795       T(J8)=(DH(J4)/2.+DH(J5))/DEL2
03800       T(J9)=DH(J5)/2./DEL2
03805    74 CONTINUE
03810       NM4=NM+4
03815       DO 75 J=1,NM4
03820 75    TZ(J)=T(J)
03825       IF(IEDGE.EQ.0) GO TO 76
03830       TZ(1)=2.-T(1)
03835       TZ(2)=2.-T(2)
03840       TZ(NM4-1)=2.-T(NM4-1)
03845       TZ(NM4)=2.-T(NM4)
03850 76    CONTINUE
03855       RETURN
03860 980   WRITE(6,981)
03865 981   FORMAT(//,' **** ERROR IN BOT INPUT')
03870       STOP
03875       END
04410C ******************************************************
04420       SUBROUTINE PLOTB(X,Y,N,NR)
04430C ******************************************************************
04440C
04450C WRITTEN 2/14/74   BY J. M. PUTNAM       DEPT 220      X23877
04460C
04470C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
04480C    N IS THE NUMBER OF POINTS TO BE PLOTTED.
04490C    NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
04500C    NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
04510C       NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
04520C
04530       REAL X(1),Y(1),HEAD(10)
04540       INTEGER  LINE(101),BLANK,STAR,PLUS
04550       DATA BLANK,STAR,PLUS /1H ,1H*,1H+/
04560       NC=51
04570       N10=(NC-1)/10
04580       WRITE(6,500)
04590 500   FORMAT(//,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
04600       WRITE(6,504)
04610       XMIN=X(1)
04620       XMAX=X(1)
04630       YMIN=Y(1)
04640       YMAX=Y(1)
```

```
C4650        DO 6 I=1,N
C4660        IF(X(I).LT.XMIN) XMIN=X(I)
C4670        IF(X(I).GT.XMAX) XMAX=X(I)
C4680        IF(Y(I).LT.YMIN) YMIN=Y(I)
C4690        IF(Y(I).GT.YMAX) YMAX=Y(I)
C4700  6     CONTINUE
C4710        DEL=XMAX-XMIN
C4720        IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C4730        XMAX=XMIN+DEL
C4740        YMAX=YMIN+DEL
C4750        DO 5 I =1,N10
C4760        Z=I
C4770     5  HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
C4780        DY=(YMAX-YMIN)/(NR-1)
C4790        IEDGE=1
C4800        IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
C4810        Z=YMAX+DY
C4820        YL=Z-DY/2.
C4830        DO 7 J=1,NR
C4840        DO 8 K=1,NC
C4850  8     LINE(K)=BLANK
C4860        Z=Z-DY
C4870        YU=YL
C4880        YL=Z-DY/2.
C4890        DO 9 I=1,N
C4900        IF(Y(I).GE.YU) GO TO 9
C4910        IF(Y(I).LT.YL) GO TO 9
C4920        K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
C4930        IF(K.GT.NC) K=NC
C4940        LINE(K)=STAR
C4950        IF(MOD(I,2).EQ.1) LINE(K)=PLUS
C4960        IF(IEDGE.EQ.0) GO TO 9
C4970        IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
C4980  9     CONTINUE
C4990        WRITE(6,508) Z,(LINE(K),K=1,NC)
C5000  7     CONTINUE
C5010        WRITE(6,504)
C5020        WRITE( 6,3002)
C5030        WRITE( 6,507) XMIN,(HEAD(I),I=1,N10)
C5040        RETURN
C5050   504  FORMAT ( 1X, 14(1H-), 1H+, 10(5H----+), 1H- )
C5060   507  FORMAT(10X,11(F10.4))
C5070   508  FORMAT (1X, F12.4,1X, 1HI, 51A1, 1HI )
C5080  3002  FORMAT(4X,7HYH / XH,4X,1HI,5(9X,1HI))
C5090        END
C5100C ------------------------------------------------------
C5102        SUBROUTINE CAPIN
C5104C
C5106C  READ CAP INPUTS AND COMPUTE CAP ARRAYS.
C5108C
C5110        COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C5112        COMMON /CAP1/ NC,XC,YC,ZC(2)
```

```
      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
      COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
      COMMON /EDG1/ NE,ZE(2),ZBE(10)
      COMMON /EDG2/ TCE(10),TPCE(10),T9E(10),TPBE(10)
      READ(5,1) NC,NPR,NE
    1 FORMAT(3I3)
      IF(NE.NE.0) NE=NC
      WRITE(6,3) NC,NPR,NE
    3 FORMAT(16H1    NC   NPR    NE,/,3I5,//)
      IF(NC.EQ.0) RETURN
      READ(5,2) XC,YC
    2 FORMAT(10F8.4)
      READ(5,2) (ZC(I),I=1,NC)
      READ(5,2) (RHOC(I),I=1,NPR)
      IF(NE.NE.0) READ(5,2) (ZE(I),I=1,NE)
      WRITE(6,4)
    4 FORMAT(37H  CAP       XC       YC       ZC       ZE,/)
      DO 100 I=1,NC
      IF(NE.EQ.0) WRITE(6,5) I,XC,YC,ZC(I)
      IF(NE.NE.0) WRITE(6,5) I,XC,YC,ZC(I),ZE(I)
  100 CONTINUE
    5 FORMAT(I4,4X,4F8.4)
      WRITE(6,6)
    6 FORMAT(//,5H RHOC)
      WRITE(6,7) (RHOC(I),I=1,NPR)
    7 FORMAT(1X,10F8.4)
      IF(MOD(NPR,2).NE.1) GO TO 980
      DO 120 I=2,NPR
  120 IF(RHOC(I).LE.RHOC(I-1)) GO TO 980
C     COMPUTATION OF CAP SECTOR PARAMETERS.
      DO 47 I=1,NP
      RC(I)=SQRT((YB(I)-YC)**2+(XB(I)-XC)**2)
   47 CONTINUE
      DO 57 I=2,NP
      I2=I-1
      RR1=YB(I2)-YC
      RR2=XB(I2)-XC
      RC1(I2)=SQRT(RR1*RR1+RR2*RR2)
      AC(I2)=ABS((XB(I)-XC)*(YB(I)+YC)+(XB(I2)-XB(I))*
     1 (YB(I2)+YB(I))+(XC-XB(I2))*(YC+YB(I2)))/2.
      SPC(I2)=RR1/RC1(I2)
      CPC(I2)=RR2/RC1(I2)
   57 CONTINUE
      DO 67 I=2,NPR
      I2=I-1
      RHOC1(I2)=(RHOC(I)+RHOC(I2))/2.
      DRHOC(I2)=RHOC(I)-RHOC(I2)
   67 CONTINUE
C     COMPUTATION OF CAP TRIANGLE FUNCTIONS.
      LC=(NPR-3)/2
      DO 74 J=1,LC
```

```
05212       J2=2*(J-1)+1
05214       J3=J2+1
05216       J4=J3+1
05218       J5=J4+1
05220       J6=4*(J-1)+1
05222       J7=J6+1
05224       J8=J7+1
05226       J9=J8+1
05228       DEL1=DRHOC(J2)+DRHOC(J3)
05230       DEL2=DRHOC(J4)+DRHOC(J5)
05232       TPCR(J6)=1./DEL1
05234       TPCR(J7)=1./DEL1
05236       TPCR(J8)=-1./DEL2
05238       TPCR(J9)=-1./DEL2
05240       TCR(J6)=DRHOC(J2)/2./DEL1
05242       TCR(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
05244       TCR(J8)=(DRHOC(J4)/2.+DRHOC(J5))/DEL2
05246       TCR(J9)=DRHOC(J5)/2./DEL2
05248    74 CONTINUE
05250       LC4=LC*4
05252       DO 75 I=1,LC4
05254       TCT(I)=TCR(I)
05256       TPCT(I)=TPCR(I)
05258 75    CONTINUE
05260       TCT(LC4-1)=2.-TCT(LC4-1)
05262       TCT(LC4)=2.-TCT(LC4)
05264       TPCT(LC4-1)=-TPCT(LC4-1)
05266       TPCT(LC4)=-TPCT(LC4)
05268       IF(RHOC(1).EQ.0.0) GO TO 76
05270       TCT(1)=2.-TCT(1)
05272       TCT(2)=2.-TCT(2)
05274       TPCT(1)=-TPCT(1)
05276       TPCT(2)=-TPCT(2)
05278 76    CONTINUE
05279       IF(NE.EQ.0) RETURN
05280C COMPUTATION OF EDGE HALF TRIANGLE FUNCTIONS.
05282       DO 80 IC=1,NC
05284       J2=NPR-2
05286       J3=J2+1
05288       J6=2*(IC-1)+1
05290       J7=J6+1
05292       DEL1=DRHOC(J2)+DRHOC(J3)
05294       TPCE(J6)=1./DEL1
05296       TPCE(J7)=1./DEL1
05298       TCE(J6)=DRHOC(J2)/2./DEL1
05300       TCE(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
05302       DEL2=ZE(IC)-ZC(IC)
05304       ZBE(J6)=ZC(IC)+0.25*DEL2
05306       ZBE(J7)=ZC(IC)+0.75*DEL2
05308       IF(DEL2.LT.0.0) GO TO 78
05310C EDGE IS AT Z=-L.
05312       TPBE(J6)=-1./DEL2
```

```
      TPBE(J7)=-1./DEL2
      TBE(J6)=0.75
      TBE(J7)=0.25
      GO TO 80
78    CONTINUE
C EDGE IS AT Z=+L.
      TPBE(J6)=1./DEL2
      TPBE(J7)=1./DEL2
      TBE(J6)=-0.75
      TBE(J7)=-0.25
80    CONTINUE
      RETURN
980   WRITE(6,981)
981   FORMAT(//,' **** ERROR IN CAP INPUT')
      STOP
      END
C-----------------------------------------------------
      SUBROUTINE WIREIN
C
C READ WIRE COORDINATES AND COMPUTE WIRE SEGMENT ARRAYS.
C
      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
     1 XW1(100),YW1(100),ZW1(100)
      COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
      COMMON /WIRE3/ NW,INDW(6),RADW(5)
      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
      COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
      NW=0
      NJ=0
      LW=0
      READ(5,51) NW,NPW,NJ
51    FORMAT(3I3)
      WRITE(6,1) NW,NPW,NJ
1     FORMAT(24H1          NW      NPW      NJ,/,3I8)
      IF(NW.EQ.0) RETURN
      READ(5,53) (XW(I),I=1,NPW)
      READ(5,53) (YW(I),I=1,NPW)
      READ(5,53) (ZW(I),I=1,NPW)
53    FORMAT(10F8.4)
      READ(5,52) (INDW(I),I=1,NW)
52    FORMAT(10I8)
      INDW(NW+1)=NPW+1
      READ(5,53) (RADW(I),I=1,NW)
      IF(NJ.EQ.0) GO TO 50
C NOTE, INDJW MUST BE MONOTONIC INCREASING.
      READ(5,52) (INDJW(I),I=1,NJ)
      READ(5,53) (RADD(I),I=1,NJ)
```

```
      READ(5,53) (UXJ(I),I=1,NJ)
      READ(5,53) (UYJ(I),I=1,NJ)
      READ(5,53) (UZJ(I),I=1,NJ)
50    IERW=0
      IERJ=0
      WRITE(6,61)
61    FORMAT(/,22X,'WIRE COORDINATES',20X,'JUNCTION PARAMETERS',//,
     1 26X,'IW',4X,'XW',6X,'YW',6X,'ZW',
     2 10X,'IJ',4X,'RADD',5X,'UXJ',5X,'UYJ',5X,'UZJ')
      IJ=1
C THIS LOOP LISTS WIRE/JUNCTION POINTS, WHILE CHECKING THE FOLLOWING:
C   1) EACH WIRE MUST CONTAIN AN ODD NUMBER OF POINTS.
C   2) EACH JUNCTION MUST EITHER START OR TERMINATE A WIRE.
C   3) CHECK THAT ALL JUNCTION POINTS ARE FOUND.
      DO 100 IW=1,NW
      WRITE(6,62) IW,RADW(IW)
62    FORMAT(2X,'WIRE',I3,' RADW=',F8.4)
      I1=INDW(IW)
      I2=INDW(IW+1)-1
C CHECK FOR AN ODD NUMBER OF POINTS ON WIRE IW.
      IF(MOD(I2-I1+1,2).NE.1) IERW=1
      DO 90 I=I1,I2
      WRITE(6,63) I,XW(I),YW(I),ZW(I)
63    FORMAT(25X,I3,3F8.4)
C CHECK IF WIRE POINT I IS A JUNCTION POINT.
      IF(IJ.GT.NJ) GO TO 90
      IF(INDJW(IJ).NE.I) GO TO 90
C CHECK THAT JUNCTION POINT IJ IS AT THE START OR END OF WIRE.
      IF(I.NE.I1 .AND. I.NE.I2) IERJ=1
      WRITE(6,64) IJ,RADD(IJ),UXJ(IJ),UYJ(IJ),UZJ(IJ)
64    FORMAT('+',56X,I3,4F8.4)
C DETERMINE DIRECTION IN WHICH WIRE LEAVES JUNCTION POINT.
      INDTJ(IJ)=1
      IF(I.EQ.I2) INDTJ(IJ)=-1
C COMPUTE JUNCTION PARAMETERS.
      RADJ(IJ)=RADW(IW)
      XJ(IJ)=XW(I)
      YJ(IJ)=YW(I)
      ZJ(IJ)=ZW(I)
      IJ=IJ+1
90    CONTINUE
100   CONTINUE
C CHECK FOR WIRE OR JUNCTION INPUT ERRORS.
      IF(IERW.NE.0) GO TO 980
      IF(IJ-1.NE.NJ .OR. IERJ.NE.0) GO TO 990
C COMPUTATION OF WIRE SEGMENT PARAMETERS.
      DO 57 I=2,NPW
      I2=I-1
      DXW(I2)=XW(I)-XW(I2)
      DYW(I2)=YW(I)-YW(I2)
      DZW(I2)=ZW(I)-ZW(I2)
      DHW(I2)=SQRT(DXW(I2)**2+DYW(I2)**2+DZW(I2)**2)
```

```
C5950       XW1(I2)=0.5*(XW(I)+XW(I2))
C5955       YW1(I2)=0.5*(YW(I)+YW(I2))
C5960       ZW1(I2)=0.5*(ZW(I)+ZW(I2))
C5965    57 CONTINUE
C5970C COMPUTATION OF WIRE TRIANGLE FUNCTIONS TW.
C5975       LW=0
C5980       DO 75 IW=1,NW
C5985       I1=INDW(IW)
C5990       I2=INDW(IW+1)-1
C5995       LW1=(I2-I1-2)/2
C6000       DO 74 J=1,LW1
C6005       LW=LW+1
C6010       J2=2*(J-1)+I1
C6015       J3=J2+1
C6020       J4=J3+1
C6025       J5=J4+1
C6030       J6=4*(LW-1)+1
C6035       J7=J6+1
C6040       J8=J7+1
C6045       J9=J8+1
C6050       INDTW(LW)=J2
C6055       DEL1=DHW(J2)+DHW(J3)
C6060       DEL2=DHW(J4)+DHW(J5)
C6065       TPW(J6)=1./DEL1
C6070       TPW(J7)=1./DEL1
C6075       TPW(J8)=-1./DEL2
C6080       TPW(J9)=-1./DEL2
C6085       TW(J6)=DHW(J2)/2./DEL1
C6090       TW(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C6095       TW(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C6100       TW(J9)=DHW(J5)/2./DEL2
C6105 74    CONTINUE
C6110 75    CONTINUE
C6115C NOTE, LW=(NPW-NW)/2-NW
C6120       IF(NJ.EQ.0) RETURN
C6125C COMPUTATION OF JUNCTION HALF TRIANGLE FUNCTIONS TJ.
C6130       DO 85 IJ=1,NJ
C6135       IF(INDTJ(IJ).GT.0) GO TO 80
C6140C JUNCTION IS AT THE END OF A WIRE.
C6145       J2=INDJW(IJ)-2
C6150       J3=J2+1
C6155       J6=2*(IJ-1)+1
C6160       J7=J6+1
C6165       INDTJ(IJ)=J2
C6170       DEL1=-DHW(J2)-DHW(J3)
C6175       TPJ(J6)=1./DEL1
C6180       TPJ(J7)=1./DEL1
C6185       TJ(J6)=DHW(J2)/2./DEL1
C6190       TJ(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C6195       GO TO 85
C6200C JUNCTION IS AT THE START OF A WIRE.
C6205 80    J4=INDJW(IJ)
```

```
06210       J5=J4+1
06215       J8=2*(IJ-1)+1
06220       J9=J8+1
06223       INDTJ(IJ)=J4
06230       DEL2=DHW(J4)+DHW(J5)
06235       TPJ(J8)=-1./DEL2
06240       TPJ(J9)=-1./DEL2
06245       TJ(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
06250       TJ(J9)=DHW(J5)/2./DEL2
06255 85    CONTINUE
06260C COMPUTE UNIT VECTORS FOR JUNCTION(DISK).
06265       DO 95 IJ=1,NJ
06270C NORMAL UNIT VECTOR (UJ).
06275       RR=SQRT(UXJ(IJ)**2+UYJ(IJ)**2+UZJ(IJ)**2)
06280       UXJ(IJ)=UXJ(IJ)/RR
06285       UYJ(IJ)=UYJ(IJ)/RR
06290       UZJ(IJ)=UZJ(IJ)/RR
06295C FIND 2 ORTHOGONAL UNIT VECTORS IN THE PLANE OF THE DISK (UJ1 & UJ2).
06300       UXJ1(IJ)=0.0
06305       UYJ1(IJ)=0.0
06310       UZJ1(IJ)=0.0
06315       UXJ2(IJ)=0.0
06320       UYJ2(IJ)=0.0
06325       UZJ2(IJ)=0.0
06330       IF(UXJ(IJ).EQ.0.0) GO TO 91
06335C FIND INTERSECTION WITH X-Y PLANE.
06340       UYJ1(IJ)=1.0
06345       UXJ1(IJ)=-UYJ(IJ)/UXJ(IJ)
06350C FIND INTERSECTION WITH X-Z PLANE.
06355       UZJ2(IJ)=1.0
06360       UXJ2(IJ)=-UZJ(IJ)/UXJ(IJ)
06365       GO TO 94
06370 91    IF(UYJ(IJ).EQ.0.0) GO TO 92
06375C FIND INTERSECTION WITH Y-Z PLANE.
06380       UZJ1(IJ)=1.0
06385       UYJ1(IJ)=-UZJ(IJ)/UYJ(IJ)
06390C FIND INTERSECTION WITH X-Y PLANE.
06395       UXJ2(IJ)=1.0
06400       UYJ2(IJ)=-UXJ(IJ)/UYJ(IJ)
06405       GO TO 94
06410 92    IF(UZJ(IJ).EQ.0.0) GO TO 94
06415C FIND INTERSECTION WITH X-Z PLANE.
06420       UZJ1(IJ)=1.0
06425       UZJ1(IJ)=-UXJ(IJ)/UZJ(IJ)
06430C FIND INTERSECTION WITH Y-Z PLANE.
06435       UYJ2(IJ)=1.0
06440       UZJ2(IJ)=-UYJ(IJ)/UZJ(IJ)
06445 94    CONTINUE
06450       RR=SQRT(UXJ1(IJ)**2+UYJ1(IJ)**2+UZJ1(IJ)**2)
06455       UXJ1(IJ)=UXJ1(IJ)/RR
06460       UYJ1(IJ)=UYJ1(IJ)/RR
06465       UZJ1(IJ)=UZJ1(IJ)/RR
```

```
06470       RR=SQRT(UXJ2(IJ)**2+UYJ2(IJ)**2+UZJ2(IJ)**2)
06475       UXJ2(IJ)=UXJ2(IJ)/RR
06480       UYJ2(IJ)=UYJ2(IJ)/RR
06485       UZJ2(IJ)=UZJ2(IJ)/RR
06490  95   CONTINUE
06495       RETURN
06500  980  WRITE(6,981)
06505  981  FORMAT(//,' **** ERROR IN WIRE INPUT')
06510       STOP
06515  990  WRITE(6,991)
06520  991  FORMAT(//,' **** ERROR IN JUNCTION INPUT')
06525       STOP
06530       END
07940C ------------------------------------------------------
07950       SUBROUTINE ORDER(MORD)
07960C DETERMINE THE ORDER IN WHICH THE Y MATRIX WAS GENERATED, AND
07970C CHECK FOR DIMENSIONAL CONSISTENCY BETWEEN THE Y MATRIX DATA FILE,
07980C AND THE INPUT DATA FILE.
07990       DIMENSION MORD(2)
08000       COMMON /BOT1/ NMODE,NPT,NBAND
08010       COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
08020       COMMON /CAP1/ NC,XC,YC,ZC(2)
08025       COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
08030       COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
08032      1 XW1(100),YW1(100),ZW1(100)
08034       COMMON /WIRE3/ NW,INDW(6),RADW(5)
08040       COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
08042       COMMON /EDG1/ NE,ZE(2),ZBE(10)
08050       READ(1) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
08052       IF(NMODE.NE.NMODE1) GO TO 500
08054       IF(NP.NE.NP1) GO TO 500
08056       IF(NC.NE.NC1) GO TO 500
08058       IF(NPR.NE.NPR1) GO TO 500
08060       IF(NE.NE.NE1) GO TO 500
08062       IF(NW.NE.NW1) GO TO 500
08064       IF(NPW.NE.NPW1) GO TO 500
08066       IF(NJ.NE.NJ1) GO TO 500
08120C MORD SPECIFIES THE ORDER IN WHICH CAPS AND/OR WIRES HAVE BEEN ADDED
08130C TO THE BOT.  INDEX 1 REFERS TO CAPS AND INDEX 2 REFERS TO WIRES.
08140C   MORD(I)=0 IF I HAS NOT BEEN ADDED
08150C   MORD(I)=N IF I HAS BEEN ADDED TO THE SYSTEM MATRIX, AND
08160C   IS LOCATED AT PSEUDO MODE NUMBER N.
08170       READ(1) MORD
08220       WRITE(6,4) MORD
08230  4    FORMAT(' THE Y MATRIX CONTAINS THE FOLLOWING ADDITIONS ( 0 IF',
08240      1 ' NOT INCLUDED, OR CORRESPONDING MODE NUMBER IF PRESENT):',//,
08250      2 10X,'CAPS',I4,/,1CX,'WIRES',I3)
08260       RETURN
08270  500  WRITE(6,1)
08280  1    FORMAT(//,' *** ERROR ***   INPUT PARAMETERS DO NOT CHECK WITH',
08290      1 ' THE PARAMETERS IN THE Y MATRIX FILE',///,
08300      2 6X,'  NMODE   NP    NC   NPR    NE    NW   NPW    NJ')
```

```
      WRITE(6,2) NMODE,NP,NC,NPR,NE,NW,NPW,NJ
    2 FORMAT(' INPUT ',8I6)
      WRITE(6,3) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
    3 FORMAT(' Y FILE',8I6)
      STOP
      END
C*****************************************************************
      SUBROUTINE RBOT(RBT,RBP,NT,TH,PHI)
C*****************************************************************
C COMPUTE BOT TRANSFER MATRICES FOR THETA AND PHI POLARIZATION.
      COMPLEX RBT(1),RBP(1),A6,U
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
      COMMON /BOT3/ DH(82),SV(82),CV(82)
      COMMON /BOT5/ T(160),TP(160),TZ(160)
      DIMENSION TH(1),PHI(1)
      DATA PI,U /3.14159265,(0.,1.)/
      DATA DTOR /0.0174533292/
      LS=NP-3
      NM=LS/2
      DO 156 K=1,NT
      K1=(K-1)*LS
      CS=COS(TH(K)*DTOR)
      SN=SIN(TH(K)*DTOR)
      CSPHI=COS(PHI(K)*DTOR)
      SNPHI=SIN(PHI(K)*DTOR)
      DO 300 J=1,NM
      J1=J+K1
      J2=J1+NM
      RBT(J1)=0.
      RBT(J2)=0.
      RBP(J1)=0.
      RBP(J2)=0.
C CALCULATION OF TRANSFER MATRICES (EQS. 38-41).
C NOTE, ONLY THE MODE INDEPENDENT PORTION IS COMPUTED HERE.
      DO 301 I=1,4
      I1=2*(J-1)+I
      I4=4*(J-1)+I
      A6=CEXP(U*BK*SN*(XB1(I1)*CSPHI+YB1(I1)*SNPHI))
C THETA POLARIZED (EQS. 38-39).
      RBT(J1)=RBT(J1)+DH(I1)*T(I4)*(CV(I1)*CSPHI+SV(I1)*SNPHI)*A6
      RBT(J2)=RBT(J2)+DH(I1)*TZ(I4)*A6
C PHI POLARIZED (EQS. 40-41).
      RBP(J1)=RBP(J1)+DH(I1)*T(I4)*(SV(I1)*CSPHI-CV(I1)*SNPHI)*A6
  301 CONTINUE
      RBT(J1)=CS*RBT(J1)
      RBT(J2)=-SN*RBT(J2)
  300 CONTINUE
  156 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------
```

```
      SUBROUTINE RCAP(RCT,RCP,NT,TH,PHI)
C
C COMPUTE CAP TRANSFER MATRICES FOR THETA AND PHI POLARIZATION.
C
      COMPLEX RCT(1),RCP(1),A6,U
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
      COMMON /BOT3/ DH(82),SV(82),CV(82)
      COMMON /BOT5/ T(160),TP(160),TZ(160)
      COMMON /CAP1/ NC,XC,YC,ZC(2)
      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
      COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
      COMMON /EDG1/ NE,ZE(2),ZBE(10)
      COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
      DIMENSION TH(1),PHI(1)
      DATA PI,U /3.14159265,(0.,1.)/
      DATA DTOR /0.017453292/
      NM=(NP-3)/2
      LR=(NPR-3)/2
      LC=NC*NM*LR
      LE=NE*NM
      DO 156 K=1,NT
      K1=(K-1)*(2*LC+LE)
      CS=COS(TH(K)*DTOR)
      SN=SIN(TH(K)*DTOR)
      CSPHI=COS(PHI(K)*DTOR)
      SNPHI=SIN(PHI(K)*DTOR)
      J1=K1
      DO 300 JC=1,NC
      DO 300 JP=1,NM
      DO 300 JR=1,LR
      J1=J1+1
      J2=J1+LC
      RCT(J1)=0.
      RCT(J2)=0.
      RCP(J1)=0.
      RCP(J2)=0.
C CALCULATION OF TRANSFER MATRICES.
      DO 301 IIP=1,4
      IP1=2*(JP-1)+IIP
      IP4=4*(JP-1)+IIP
      DO 301 IIR=1,4
      IR1=2*(JR-1)+IIR
      IR4=4*(JR-1)+IIR
      XX=XC+RHOC1(IR1)*RC1(IP1)*CPC(IP1)
      YY=YC+RHOC1(IR1)*RC1(IP1)*SPC(IP1)
      ZZ=ZC(JC)
      A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
      AI=AC(IP1)*ABS(RHOC(IR1+1)**2-RHOC(IR1)**2)
C THETA POLARIZED.
      RCT(J1)=RCT(J1)+AI*(CV(IP1)*CSPHI+SV(IP1)*SNPHI)*
```

```
     1  T(IP4)*TCT(IR4)*A6
      RCT(J2)=RCT(J2)+AI*(CPC(IP1)*CSPHI+SPC(IP1)*SNPHI)*
     1  TZ(IP4)*TCR(IR4)*A6
C     PHI POLARIZED.
      RCP(J1)=RCP(J1)+AI*(SV(IP1)*CSPHI-CV(IP1)*SNPHI)*
     1  T(IP4)*TCT(IR4)*A6
      RCP(J2)=RCP(J2)+AI*(SPC(IP1)*CSPHI-CPC(IP1)*SNPHI)*
     1  TZ(IP4)*TCR(IR4)*A6
  301 CONTINUE
      RCT(J1)=CS*RCT(J1)
      RCT(J2)=CS*RCT(J2)
  300 CONTINUE
      IF(NE.EQ.0) GO TO 156
C     EDGE TRANSFER MATRICES FOLLOW.
      J2=2*LC+K1
      DO 400 JC=1,NC
      DO 400 JP=1,NM
      J2=J2+1
      RCT(J2)=0.
      RCP(J2)=0.
      DO 401 IIP=1,4
      IP1=2*(JP-1)+IIP
      IP4=4*(JP-1)+IIP
      DO 401 IIRZ=1,2
      IR1=2*LR+IIRZ
      IR4=2*(JC-1)+IIRZ
      IZ1=2*(JC-1)+IIRZ
      IZ4=2*(JC-1)+IIRZ
      XX=XC+RHOC1(IR1)*RC1(IP1)*CPC(IP1)
      YY=YC+RHOC1(IR1)*RC1(IP1)*SPC(IP1)
      ZZ=ZC(JC)
      A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
      AI=AC(IP1)*ABS(RHOC(IR1+1)**2-RHOC(IR1)**2)
C     EDGE(CAP)
C     THETA POLARIZED.
      RCT(J2)=RCT(J2)+CS*AI*(CPC(IP1)*CSPHI+SPC(IP1)*SNPHI)*
     1  TZ(IP4)*TCE(IR4)*A6
C     PHI POLARIZED.
      RCP(J2)=RCP(J2)+AI*(SPC(IP1)*CSPHI-CPC(IP1)*SNPHI)*
     1  TZ(IP4)*TCE(IR4)*A6
      XX=XB1(IP1)
      YY=YB1(IP1)
      ZZ=ZBE(IZ1)
      A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
      AI=DH(IP1)*ABS(ZE(JC)-ZC(JC))/2.
C     EDGE(BOT).
C     THETA POLARIZED.
      RCT(J2)=RCT(J2)-SN*AI*TZ(IP4)*TBE(IZ4)*A6
  401 CONTINUE
  400 CONTINUE
  156 CONTINUE
      RETURN
```

```
09216       END
C9217C ------------------------------------------------------
09218       SUBROUTINE RWIRE(RWT,RWP,NT,TH,PHI)
09219C
09220C  COMPUTE WIRE-JUNCTION TRANSFER MATRICES.
09230C
09240       COMPLEX RWT(1),RWP(1),A6,U
09245       COMMON /WAVE/ BK
09250       COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
09260      1 XW1(100),YW1(100),ZW1(100)
09270       COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
09280       COMMON /WIRE3/ NW,INDW(6),RADW(5)
09290       COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
09300       COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
09310       COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
09320       COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
09324       COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
09326       COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
09340       DIMENSION TH(1),PHI(1)
09350       DATA PI,U /3.14159265,(0.,1.)/
09360       DATA DTOR /0.017453292/
09380       LWJ=LW+NJ
09390       DO 156 K=1,NT
09400       K1=(K-1)*LWJ
09410       CS=COS(TH(K)*DTOR)
09420       SN=SIN(TH(K)*DTOR)
09430       CSPHI=COS(PHI(K)*DTOR)
09440       SNPHI=SIN(PHI(K)*DTOR)
09450       DO 300 J=1,LW
09455       J1=J+K1
09460       RWT(J1)=0.
09465       RWP(J1)=0.
09470       DO 301 I=1,4
09475       I1=INDTW(J)+I-1
09480       I4=4*(J-1)+I
09485       A6=CEXP(U*BK*(SN*(XW1(I1)*CSPHI+YW1(I1)*SNPHI)+ZW1(I1)*CS))
09490       RWT(J1)=RWT(J1)+(CS*(DXW(I1)*CSPHI+DYW(I1)*SNPHI)-
09495      1 DZW(I1)*SN)*TW(I4)*A6
09500       RWP(J1)=RWP(J1)+(-DXW(I1)*SNPHI+DYW(I1)*CSPHI)*TW(I4)*A6
09505  301  CONTINUE
09510  300  CONTINUE
09515       IF(NJ.EQ.0) GO TO 156
09520       J1=LW+K1
09525       DO 400 J=1,NJ
09530       J1=J1+1
09535       RWT(J1)=0.
09540       RWP(J1)=0.
09545       DO 401 I=1,2
09550       I1=INDTJ(J)+I-1
09555       I2=2*(J-1)+I
09560       A6=CEXP(U*BK*(SN*(XW1(I1)*CSPHI+YW1(I1)*SNPHI)+ZW1(I1)*CS))
09565       RWT(J1)=RWT(J1)+(CS*(DXW(I1)*CSPHI+DYW(I1)*SNPHI)-
```

```
09570      1 DZW(I1)*SN)*TJ(I2)*A6
09575        RWP(J1)=RWP(J1)+(-DXW(I1)*SNPHI+DYW(I1)*CSPHI)*TJ(I2)*A6
09580  401   CONTINUE
09585        DO 402 JJ=1,4
09590        ALPHA=PI*((JJ-1)/2.+0.25)
09595        SA=SIN(ALPHA)
09600        CA=COS(ALPHA)
09605        RA=(RADJ(J)+RADD(J))/2.
09610        XA=XJ(J)+RA*(CA*UXJ1(J)+SA*UXJ2(J))
09612        YA=YJ(J)+RA*(CA*UYJ1(J)+SA*UYJ2(J))
09614        ZA=ZJ(J)+RA*(CA*UZJ1(J)+SA*UZJ2(J))
09616        DT=CA*(UXJ1(J)*CS*CSPHI+UYJ1(J)*CS*SNPHI-UZJ1(J)*SN)+
09618      1 SA*(UXJ2(J)*CS*CSPHI+UYJ2(J)*CS*SNPHI-UZJ2(J)*SN)
09620        DP=CA*(-UXJ1(J)*SNPHI+UYJ1(J)*CSPHI)+
09622      1 SA*(-UXJ2(J)*SNPHI+UYJ2(J)*CSPHI)
09625        RR=(RADJ(J)-RADD(J))/2.
09630        A6=CEXP(U*BK*(SN*(XA*CSPHI+YA*SNPHI)+ZA*CS))
09635        RWT(J1)=RWT(J1)+RP/4.*DT*A6
09645        RWP(J1)=RWP(J1)+RR/4.*DP*A6
09650  402   CONTINUE
09660  400   CONTINUE
09670   156  CONTINUE
09680        RETURN
09690        END
09700C *****************************************************************
09702        SUBROUTINE LPCUR(NMODE,NM,CB,LW,NJ,CW,LC,LE,CC)
09704C *****************************************************************
09706C LIST AND PLOT BOT, WIRE, AND CAP CURRENTS.
09708        COMPLEX U,A3,A4,CB(1),CW(1),CC(1)
09710        DIMENSION TMAG(41),TPHS(41),ZMAG(41),ZPHS(41)
09712        DATA PI,U /3.14159265,(0.,1.)/
09714        WRITE(6,3)
09716  3     FORMAT(16H1    BOT CURRENTS)
09718        KM=2*NMODE-1
09720        LS=NM*2
09722        DO 100 MM=1,KM
09724        M=MM-NMODE
09726        I1=(MM-1)*LS+1
09728        I2=I1+NM-1
09730        WRITE(6,1) M,(CB(I),I=I1,I2)
09732  1     FORMAT(/,29H T-DIRECTED CURRENTS FOR MODE,I3,/,4(2E12.4,5X))
09734        I1=I2+1
09736        I2=I1+NM-1
09738        WRITE(6,2) M,(CB(I),I=I1,I2)
09740  2     FORMAT(/,29H Z-DIRECTED CURRENTS FOR MODE,I3,/,4(2E12.4,5X))
09742  100   CONTINUE
09744        DO 320 I=1,NM
09746        DO 310 K=1,41
09748        A3=0.0
09750        A4=0.0
09752        ZE=-1.0+(K-1)*0.05
09754        M=-NMODE
```

```
09756       I1=I-LS
09758       DO 305 MM=1,KM
09760       M=M+1
09762       I1=I1+LS
09764       I2=I1+NM
09766C SUM THE T AND Z DIRECTED CURRENTS ON TRIANGLE FUNCTION I AT Z=ZE,
09768C (EQ. 8).
09770       A3=A3+CB(I1)*CEXP(U*M*PI*ZE)
09772       A4=A4+CB(I2)*(CEXP(U*M*PI*ZE)-(-1)**M)
09774 305   CONTINUE
09776       TMAG(K)=CABS(A3)/2.
09778       TPHS(K)=ATAN2(AIMAG(A3),REAL(A3))*180./PI
09780       ZMAG(K)=CABS(A4)/2.
09782       ZPHS(K)=ATAN2(AIMAG(A4),REAL(A4))*180./PI
09784 310   CONTINUE
09786       WRITE(6,301)
09788 301   FORMAT(1H1)
09790       WRITE(6,302) I
09792 302   FORMAT(41H T-DIRECTED CURRENTS ON TRIANGLE FUNCTION,I3,/)
09794C PLOT MAGNITUDE AND PHASE.
09796       CALL PLOTC(TMAG,TPHS)
09798       WRITE(6,304) I
09800 304   FORMAT(///,41H Z-DIRECTED CURRENTS ON TRIANGLE FUNCTION,I3,/)
09802C PLOT MAGNITUDE AND PHASE.
09804       CALL PLOTC(ZMAG,ZPHS)
09806 320   CONTINUE
09808       IF(LW.EQ.0) GO TO 500
09810       WRITE(6,4)
09812 4     FORMAT(//,16H   WIRE CURRENTS,/)
09814       WRITE(6,5) (CW(I),I=1,LW)
09816 5     FORMAT(4(2E12.4,5X))
09818 500   CONTINUE
09820       IF(NJ.EQ.0) GO TO 600
09822       I1=LW+1
09824       I2=LW+NJ
09826       WRITE(6,6)
09828 6     FORMAT(//,20H   JUNCTION CURRENTS,/)
09830       WRITE(6,5) (CW(I),I=I1,I2)
09832 600   CONTINUE
09834       IF(LC.EQ.0) RETURN
09836       WRITE(6,7)
09838 7     FORMAT(//,15H   CAP CURRENTS,/)
09840       WRITE(6,8)
09842 8     FORMAT(/,11H T-DIRECTED)
09844       WRITE(6,5) (CC(I),I=1,LC)
09846       WRITE(6,9)
09848 9     FORMAT(/,11H R-DIRECTED)
09850       I1=LC+1
09852       I2=2*LC
09854       WRITE(6,5) (CC(I),I=I1,I2)
09856       IF(LE.EQ.0) RETURN
09858       WRITE(6,10)
```

```
09860 10    FORMAT(///,14H EDGE CURRENTS)
09862       I1=2*LC+1
09864       I2=2*LC+LE
09866       WRITE(6,5) (CC(I),I=I1,I2)
09868       RETURN
09870       END
10270C ********************************************************************
10280       SUBROUTINE PLOTC(Y1,Y2)
10290C ********************************************************************
10300C
10310C WRITTEN 2/14/74   BY J. M. PUTNAM       DEPT 220     X23877
10320C
10330C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
10340C    N IS THE NUMBER OF POINTS TO BE PLOTTED.
10350C    NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
10360C    NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
10370C       NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
10380C
10390       REAL Y1(41),Y2(41),HEAD(10)
10400       INTEGER  LINE(101),BLANK,STAR,PLUS
10410       DATA BLANK,STAR,PLUS /1H ,1H*,1H+/
10420       N=41
10430       NR=21
10440       NC=41
10450       N10=(NC-1)/10
10460       WRITE(6,500)
10470 500   FORMAT(12H * MAGNITUDE,45X,7H+ PHASE)
10480       WRITE(6,50+)
10490       XMIN=-1.0
10500       XMAX=1.0
10510       YMIN1=Y1(1)
10520       YMAX1=Y1(1)
10530       YMIN2=-180.0
10540       YMAX2=180.0
10550       DO 6 I=1,N
10560       IF(Y1(I).LT.YMIN1) YMIN1=Y1(I)
10570       IF(Y1(I).GT.YMAX1) YMAX1=Y1(I)
10580 6     CONTINUE
10590       DO 5 I =1,N10
10600       Z=I
10610     5 HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
10620       DY1=(YMAX1-YMIN1)/(NR-1)
10630       DY2=18.0
10640       Z1=YMAX1+DY1
10650       Z2=YMAX2+DY2
10660       YL1=Z1-DY1/2.
10670       YL2=Z2-DY2/2.
10680       DO 7 J=1,NR
10690       DO 8 K=1,NC
10700 8     LINE(K)=BLANK
10710       Z1=Z1-DY1
10720       Z2=Z2-DY2
```

```
      YU1=YL1
      YU2=YL2
      YL1=Z1-DY1/2.
      YL2=Z2-DY2/2.
      DO 9 I=1,N
      X=XMIN+(XMAX-XMIN)*(I-1)/FLOAT(N-1)
      IF(Y2(I).GE.YU2) GO TO 10
      IF(Y2(I).LT.YL2) GO TO 10
      K=(X-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
      IF(K.GT.NC) K=NC
      LINE(K)=PLUS
10    IF(Y1(I).GE.YU1) GO TO 9
      IF(Y1(I).LT.YL1) GO TO 9
      K=(X-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
      IF(K.GT.NC) K=NC
      LINE(K)=STAR
9     CONTINUE
      WRITE(6,508) Z1,(LINE(K),K=1,NC),Z2
7     CONTINUE
      WRITE(6,504)
      WRITE( 6,3002)
      WRITE( 6,507) XMIN,(HEAD(I),I=1,N10)
      RETURN
 504  FORMAT ( 1X, 14(1H-), 1H+, 8(5H----+), 1H- )
 507  FORMAT(3X,3HZ/L,5X,11(G10.2))
 508  FORMAT (1X, G12.3,1X, 1HI, 41A1, 1HI,F7.1)
3002  FORMAT(15X,1HI,4(9X,1HI))
      END
C-----------------------------------------------------
      SUBROUTINE NEARB(XTEST,YTEST,ZTEST,ZM)
C CALCULATE THE M-TH MODAL CURRENT COEFFICIENTS FOR THE NEAR FIELD
C ANALYSIS (EQS 67 & 71).
      COMPLEX U,A3,ZM(1)
      COMPLEX G1(82),GZ(82),G1T(82),G1Z(82),H1T(82)
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
      COMMON /BOT3/ DH(82),SV(82),CV(82)
      COMMON /BOT5/ T(160),TP(160),TZ(160)
      COMMON /INT/ M,RH(2,ZP
      EXTERNAL FUNCT,FUNCZ,FUNC1T,FUNC1Z,FUNC2T
      DATA PI,U /3.14159265,(0.,1.)/
      BKL=BK*BL
      KG=NP-1
      NM=(NP-3)/2
      ETA=376.707
      II=0
C COMPUTATION OF GREEN FUNCTION KERNELS G, H0, AND H1.
      DO 16 J=1,KG
      II=II+1
      YY=YB1(J)-YTEST
      XX=XB1(J)-XTEST
      RH02=XX*XX+YY*YY
```

```
11068       ZP=ZTEST
11070       CALL CSIMP(FUNCT,-BL,BL,0.05,10,A3,GT(I1),K,IER)
11072       GT(I1)=GT(I1)/4./PI
11074       CALL CSIMP(FUNCZ,-BL,BL,0.05,10,A3,GZ(I1),K,IER)
11076       GZ(I1)=GZ(I1)/4./PI
11078       CALL CSIMP(FUNC1T,-BL,BL,0.05,10,A3,G1T(I1),K,IER)
11080       G1T(I1)=G1T(I1)/4./PI
11082       CALL CSIMP(FUNC1Z,-BL,BL,0.05,10,A3,G1Z(I1),K,IER)
11084       G1Z(I1)=G1Z(I1)/4./PI
11086       CALL CSIMP(FUNC2T,-BL,BL,0.05,10,A3,H1T(I1),K,IER)
11088       H1T(I1)=H1T(I1)/4./PI
11090 16    CONTINUE
11092C COMPUTATION OF M-TH MODAL CURRENT COEFFICIENTS (EQS. 67 & 71).
11094       DO 3C J=1,NM
11096       JL=(J-1)*6
11098       J3=(J-1)*4
11100       J1=2*(J-1)
11102C ELECTRIC FIELD.
11104       L1=JL+1
11106       L2=L1+NM*6
11108       L3=L1+1
11110       L4=L2+1
11112       L5=L1+2
11114       L6=L2+2
11116       ZM(L1)=0.
11118       ZM(L2)=0.
11120       ZM(L3)=0.
11122       ZM(L4)=0.
11124       ZM(L5)=0.
11126       ZM(L6)=0.
11128C MAGNETIC FIELD.
11130       K1=L1+3
11132       K2=L2+3
11134       K3=L1+4
11136       K4=L2+4
11138       K5=L1+5
11140       K6=L2+5
11142       ZM(K1)=0.
11144       ZM(K2)=0.
11146       ZM(K3)=0.
11148       ZM(K4)=0.
11150       ZM(K5)=0.
11152       ZM(K6)=0.
11154       DO 7C JJ=1,4
11156       J2=J1+JJ
11158       J7=J3+JJ
11160       J4=J2
11162       XX=X81(J2)-XTEST
11164       YY=Y81(J2)-YTEST
11166C ELECTRIC FIELD.
11168C DX COMPONENT.
11170       ZM(L1)=ZM(L1)+DH(J2)*(CV(J2)*T(J7)*GT(J4)+
```

```
     1  XX*TP(J7)*G1T(J4)/BK/BK)
        ZM(L2)=ZM(L2)+DH(J2)*XX*TZ(J7)*G1T(J4)
C UY COMPONENT.
        ZM(L3)=ZM(L3)+DH(J2)*(SV(J2)*T(J7)*GT(J4)+
     1  YY*TP(J7)*G1T(J4)/BK/BK)
        ZM(L4)=ZM(L4)+DH(J2)*YY*TZ(J7)*G1T(J4)
C UZ COMPONENT.
        ZM(L5)=ZM(L5)+DH(J2)*TP(J7)*H1T(J4)
        ZM(L6)=ZM(L6)+DH(J2)*TZ(J7)*(GZ(J4)+
     1  U*M*PI*H1T(J4)/BK/BK)
C MAGNETIC FIELD.
C UX COMPONENT.
        ZM(K1)=ZM(K1)-DH(J2)*SV(J2)*T(J7)*H1T(J4)
        ZM(K2)=ZM(K2)+DH(J2)*YY*TZ(J7)*G1Z(J4)
C UY COMPONENT.
        ZM(K3)=ZM(K3)+DH(J2)*CV(J2)*T(J7)*H1T(J4)
        ZM(K4)=ZM(K4)-DH(J2)*XX*TZ(J7)*G1Z(J4)
C UZ COMPONENT.
        ZM(K5)=ZM(K5)-DH(J2)*T(J7)*(YY*CV(J2)-
     1  XX*SV(J2))*G1T(J4)
   70 CONTINUE
C MULTIPLY BY CONSTANTS.
        ZM(L1)=U*BK*ETA*ZM(L1)
        ZM(L2)=-M*PI*ETA/BKL*ZM(L2)
        ZM(L3)=U*BK*ETA*ZM(L3)
        ZM(L4)=-M*PI*ETA/BKL*ZM(L4)
        ZM(L5)=U*ETA/BK*ZM(L5)
        ZM(L6)=U*BK*ETA*ZM(L6)
   30 CONTINUE
      RETURN
      END
C-------------------------------------------------------
      SUBROUTINE NEARC(XTEST,YTEST,ZTEST,ZM)
C CALCULATE THE CAP CURRENT COEFFICIENTS FOR THE NEAR FIELD ANALYSIS.
      COMPLEX U,A3,A4,ZM(1)
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
      COMMON /BOT3/ DH(82),SV(82),CV(82)
      COMMON /BOT5/ T(160),TP(160),TZ(160)
      COMMON /CAP1/ NC,XC,YC,ZC(2)
      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
      COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
      DATA PI,U /3.14159265,(0.,1.)/
      NM=(NP-3)/2
      LR=(NPR-3)/2
      LC=NC*NM*LR
      BK2=BK*BK
      ETA=376.707
      J=0
C NOTE, EDGE COEFFICIENTS ARE NOT INCLUDED.
      DO 30 JC=1,NC
```

```
13790         DO 30 JP=1,NM
13800         DO 30 JR=1,LR
13810         J=J+1
13820         JL=(J-1)*6
13830C ELECTRIC FIELD.
13840         L1=JL+1
13850         L2=L1+LC*6
13860         L3=L1+1
13870         L4=L2+1
13880         L5=L1+2
13890         L6=L2+2
13900         ZM(L1)=0.
13910         ZM(L2)=0.
13920         ZM(L3)=0.
13930         ZM(L4)=0.
13940         ZM(L5)=0.
13950         ZM(L6)=0.
13960C MAGNETIC FIELD.
13970         K1=L1+3
13980         K2=L2+3
13990         K3=L1+4
14000         K4=L2+4
14010         K5=L1+5
14020         K6=L2+5
14030         ZM(K1)=0.
14040         ZM(K2)=0.
14050         ZM(K3)=0.
14060         ZM(K4)=0.
14070         ZM(K5)=0.
14080         ZM(K6)=0.
14090         JP2=2*(JP-1)
14100         JP7=4*(JP-1)
14110         DO 70 JJP=1,4
14120         JP2=JP2+1
14130         JP7=JP7+1
14140         JR2=2*(JR-1)
14150         JR7=4*(JR-1)
14160         DO 70 JJR=1,4
14170         JR2=JR2+1
14180         JR7=JR7+1
14190         AJ=AC(JP2)*ABS(RHOC(JR2+1)**2-RHOC(JR2)**2)
14200         RJ=RHOC1(JR2)*RC1(JP2)
14210         XX=XC+RJ*CPC(JP2)-XTEST
14220         YY=YC+RJ*SPC(JP2)-YTEST
14230         ZZ=ZC(JC)-ZTEST
14240         RR=SQRT(XX*XX+YY*YY+ZZ*ZZ)
14250         A3=CEXP(-U*BK*RR)/4./PI/RR
14260         A4=(1./RR+U*BK)/RR*A3
14270C ELECTRIC FIELD.
14280C UX COMPONENT.
14290         ZM(L1)=ZM(L1)+AJ*TCT(JR7)*(CV(JP2)*T(JP7)*A3+
14300        1  XX/BK2*TP(JP7)/RHOC1(JR2)*A4)
```

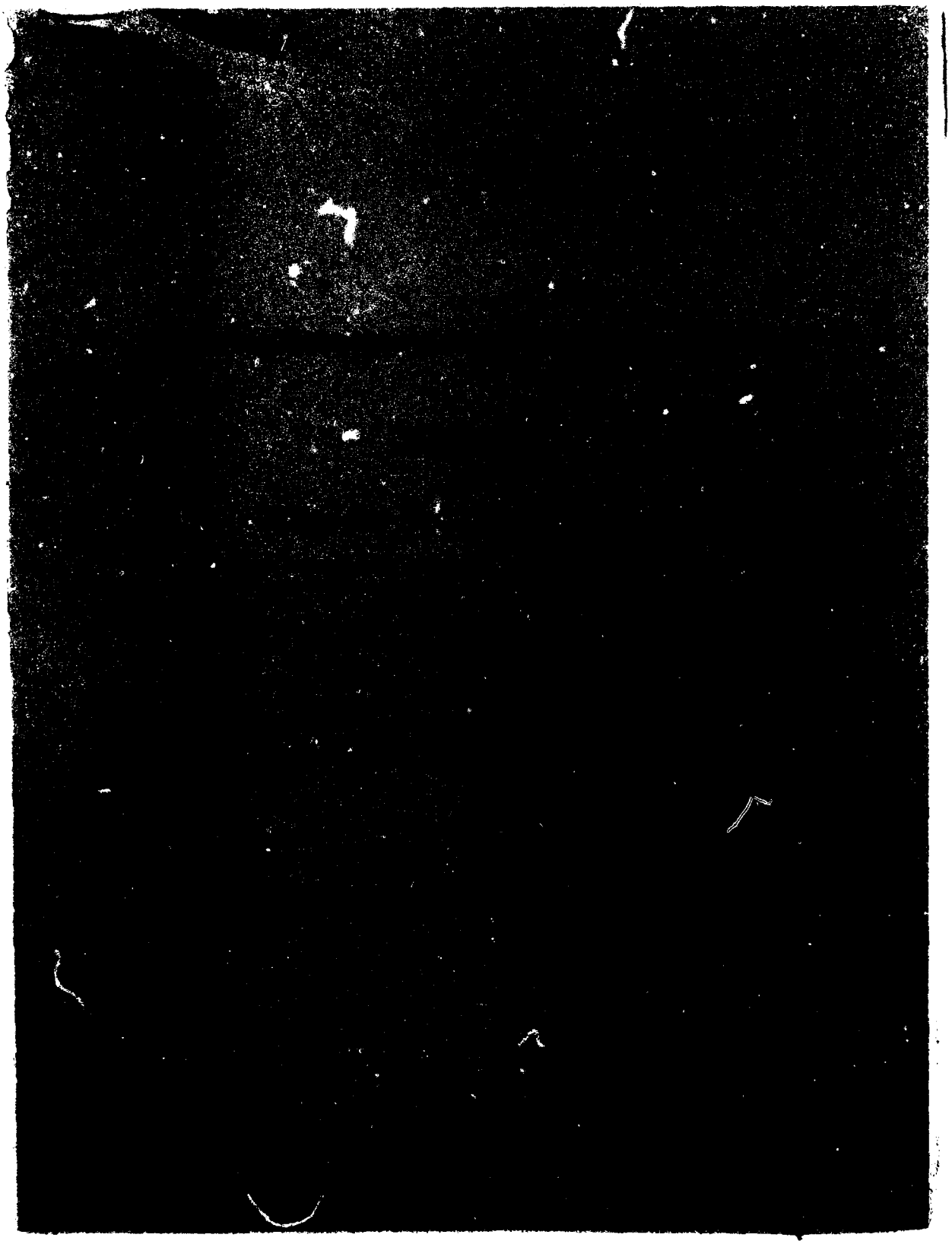
```
      ZM(L2)=ZM(L2)+AJ*TZ(JP7)*(CPC(JP2)*TCR(JR7)*A3+
     1  XX/BK2*(TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)*A4)
C UY COMPONENT.
      ZM(L3)=ZM(L3)+AJ*TCT(JR7)*(SV(JP2)*T(JP7)*A3+
     1  YY/BK2*TP(JP7)/RHOC1(JR2)*A4)
      ZM(L4)=ZM(L4)+AJ*TZ(JP7)*(SPC(JP2)*TCR(JR7)*A3+
     1  YY/BK2*(TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)*A4)
C UZ COMPONENT.
      ZM(L5)=ZM(L5)+AJ*ZZ*TP(JP7)/RHOC1(JR2)*TCT(JR7)*A4
      ZM(L6)=ZM(L6)+AJ*ZZ*TZ(JP7)*
     1  (TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)*A4
C MAGNETIC FIELD.
C UX COMPONENT.
      ZM(K1)=ZM(K1)-AJ*ZZ*SV(JP2)*T(JP7)*TCT(JR7)*A4
      ZM(K2)=ZM(K2)-AJ*ZZ*SPC(JP2)*TZ(JP7)*TCR(JR7)*A4
C UY COMPONENT.
      ZM(K3)=ZM(K3)+AJ*ZZ*CV(JP2)*T(JP7)*TCT(JR7)*A4
      ZM(K4)=ZM(K4)+AJ*ZZ*CPC(JP2)*TZ(JP7)*TCR(JR7)*A4
C UZ COMPONENT.
      ZM(K5)=ZM(K5)-AJ*T(JP7)*TCT(JR7)*
     1  (YY*CV(JP2)-XX*SV(JP2))*A4
      ZM(K6)=ZM(K6)-AJ*TZ(JP7)*TCR(JR7)*
     1  (YY*CPC(JP2)-XX*SPC(JP2))*A4
   70 CONTINUE
C MULTIPLY BY CONSTANTS.
      ZM(L1)=U*BK*ETA*ZM(L1)
      ZM(L2)=U*BK*ETA*ZM(L2)
      ZM(L3)=U*BK*ETA*ZM(L3)
      ZM(L4)=U*BK*ETA*ZM(L4)
      ZM(L5)=U*ETA/BK*ZM(L5)
      ZM(L6)=U*ETA/BK*ZM(L6)
   30 CONTINUE
      RETURN
      END
C-----------------------------------------------
      SUBROUTINE NEARW(XTEST,YTEST,ZTEST,ZM)
C CALCULATE THE WIRE/JUNCTION CURRENT COEFFICIENTS FOR THE NEAR
C FIELD ANALYSIS.
      COMPLEX U,A3,A4,ZM(1)
      COMMON /WAVE/ BK
      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
     1  XW1(100),YW1(100),ZW1(100)
      COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
      COMMON /WIRE3/ NW,INOW(6),RADW(5)
      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
      COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
      DATA PI,U /3.14159265,(0.,1.)/
```

```
      LWJ=LW+NJ
      BK2=BK*BK
      ETA=376.707
C NOTE, JUNCTION COEFFICIENTS ARE NOT INCLUDED.
      DO 30 J=1,LW
      JL=(J-1)*6
      J3=(J-1)*4
      J1=INDTW(J)-1
C ELECTRIC FIELD.
      L1=JL+1
      L3=L1+1
      L5=L1+2
      ZM(L1)=0.
      ZM(L3)=0.
      ZM(L5)=0.
C MAGNETIC FIELD.
      K1=L1+3
      K3=L1+4
      K5=L1+5
      ZM(K1)=0.
      ZM(K3)=0.
      ZM(K5)=0.
      DO 70 JJ=1,4
      J2=J1+JJ
      J7=J3+JJ
      XX=XW1(J2)-XTEST
      YY=YW1(J2)-YTEST
      ZZ=ZW1(J2)-ZTEST
      RR=SQRT(XX*XX+YY*YY+ZZ*ZZ)
      A3=CEXP(-U*BK*RR)/4./PI/RR
      A4=(1./RR+U*BK)/RR*A3
C ELECTRIC FIELD.
C UX COMPONENT.
      ZM(L1)=ZM(L1)+DXW(J2)*TW(J7)*A3+
     1 DHW(J2)*XX*TPW(J7)/BK2*A4
C UY COMPONENT.
      ZM(L3)=ZM(L3)+DYW(J2)*TW(J7)*A3+
     1 DHW(J2)*YY*TPW(J7)/BK2*A4
C UZ COMPONENT.
      ZM(L5)=ZM(L5)+DZW(J2)*TW(J7)*A3+
     1 DHW(J2)*ZZ*TPW(J7)/BK2*A4
C MAGNETIC FIELD.
C UX COMPONENT.
      ZM(K1)=ZM(K1)-(DYW(J2)*ZZ-
     1 DZW(J2)*YY)*TW(J7)*A4
C UY COMPONENT.
      ZM(K3)=ZM(K3)-(DZW(J2)*XX-
     1 DXW(J2)*ZZ)*TW(J7)*A4
C UZ COMPONENT.
      ZM(K5)=ZM(K5)-(DXW(J2)*YY-
     1 DYW(J2)*XX)*TW(J7)*A4
   70 CONTINUE
```

```
C MULTIPLY BY CONSTANTS.
      ZM(L1)=U*BK*ETA*ZM(L1)
      ZM(L3)=U*BK*ETA*ZM(L3)
      ZM(L5)=U*BK*ETA*ZM(L5)
   30 CONTINUE
      RETURN
      END
C*******************************************************************
      SUBROUTINE CSIMP(F,A,B,DEL,IMAX,SI1,S,N,IER)
C*******************************************************************
C COMPLEX SIMPSON INTEGRATION ROUTINE.
      COMPLEX F,SI1,S,SUMK
      SI1=(.0,.0)
      S=(.0,.0)
      N=0
      BA=B-A
      IF (BA)20,19,20
   19 IER=1
      RETURN
   20 IF(DEL)22,22,23
   22 IER=2
      RETURN
   23 IF(IMAX-1)24,24,25
   24 IER=3
      RETURN
   25 X=BA/2.+A
      NHALF=1
      SUMK=F(X)*BA*2./3.
      S=SUMK+(F(A)+F(B))*BA/6.
      DO 28 I=2,IMAX
      SI1=S
      S=(S-SUMK/2.)/2.
      NHALF=NHALF*2
      ANHLF=NHALF
      FRSTX=A+(BA/ANHLF)/2.
      SUMK=F(FRSTX)
      XK=FRSTX
      KLAST=NHALF-1
      FINC=BA/ANHLF
      DO 26 K=1,KLAST
      XK=XK+FINC
   26 SUMK=SUMK+F(XK)
      SUMK=SUMK*2.*BA/(3.*ANHLF)
      S=S+SUMK
      IF(CABS(S).EQ.0.0) GO TO 29
      IF((CABS(S-SI1)/CABS(S))-DEL) 29,28,28
   28 CONTINUE
      IER=4
      WRITE(6,1) SI1,S
    1 FORMAT(//,29H INTEGRATION DID NOT CONVERGE,//
     1 50H THE PREVIOUS AND FINAL VALUES OF THE INTEGRAL ARE,
     2 13H RESPECTIVELY,4E12.3)
```

```
      GO TO 30
   29 IER=0
   30 N=2*NHALF
      RETURN
      END
C -------------------------------------------------
      COMPLEX FUNCTION FUNCT(Z)
      COMPLEX U
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL
      COMMON /INT/ M,RHO2,ZP
      DATA PI,U /3.14159265,(0.,1.)/
      R=SQRT(RHO2+(Z-ZP)**2)
      FUNCT=CEXP(U*(M*PI*Z/BL-BK*R))/R
      RETURN
      END
C -------------------------------------------------
      COMPLEX FUNCTION FUNCZ(Z)
      COMPLEX U
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL
      COMMON /INT/ M,RHO2,ZP
      DATA PI,U /3.14159265,(0.,1.)/
      R=SQRT(RHO2+(Z-ZP)**2)
      FUNCZ=(CEXP(U*M*PI*Z/BL)-(-1)**M)*CEXP(-U*BK*R)/R
      RETURN
      END
C -------------------------------------------------
      COMPLEX FUNCTION FUNC1T(Z)
      COMPLEX U
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL
      COMMON /INT/ M,RHO2,ZP
      DATA PI,U /3.14159265,(0.,1.)/
      R=SQRT(RHO2+(Z-ZP)**2)
      FUNC1T=CEXP(U*(M*PI*Z/BL-BK*R))*(1./R+U*BK)/R/R
      RETURN
      END
C -------------------------------------------------
      COMPLEX FUNCTION FUNC1Z(Z)
      COMPLEX U
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL
      COMMON /INT/ M,RHO2,ZP
      DATA PI,U /3.14159265,(0.,1.)/
      R=SQRT(RHO2+(Z-ZP)**2)
      FUNC1Z=(CEXP(U*M*PI*Z/BL)-(-1)**M)*CEXP(-U*BK*R)*
     1 (1./R+U*BK)/R/R
      RETURN
      END
C -------------------------------------------------
      COMPLEX FUNCTION FUNC2T(Z)
```

```
      COMPLEX U
      COMMON /WAVE/ BK
      COMMON /BOT2/ NP,BL
      COMMON /INT/ M,RHO2,ZP
      DATA PI,U /3.14159265,(0.,1.)/
      R=SQRT(RHO2+(Z-ZP)**2)
      FUNC2T=CEXP(U*(M*PI*Z/BL-BK*R))*(Z-ZP)*(1./R+U*BK)/R/R
      RETURN
      END
/EOR
```



END

DATE
FILMED

9 82

DTIC